Impact des liens hypertextes sur la précision en recherche d'information.

Idir Chibane

N° d'ordre : 9308

UNIVERSITÉ PARIS SUD

UFR SCIENTIFIQUE D'ORSAY

THÈSE

Présentée pour obtenir le grade de

DOCTEUR EN SCIENCES

DE

L'UNIVERSITÉ PARIS XI ORSAY

Spécialité : INFORMATIQUE
Préparée au Département Informatique de Supélec à Gif-sur-Yvette

par

Idir CHIBANE

SUJET :

Impact des liens hypertextes sur la précision en recherche d'information.

Conception d'un système de recherche d'information adapté au Web

Soutenue publiquement le 10 décembre 2008 devant la commission d'examen

M. Mohand BOUGHANEM, Professeur, Université Paul Sabatier (Rapporteur)
Mme Bich-Liên DOAN, Enseignant-Chercheur, Supélec (Encadrante)
M. Gilles FALQUET, Professeur, Université de Genève (Rapporteur)
M. Christian JACQUEMIN, Professeur, CNRS-LIMSI & Université Paris 11 (Directeur de thèse)
Mme Anne VILNAT, Professeur, CNRS-LIMSI & Université Paris 11 (Examinateur)

Remerciements

C'est avec une immense joie que je formule ces remerciements qui témoignent par écrit ma reconnaissance à toutes les personnes qui ont de près ou de loin manifesté leur soutien et leur confiance en ma personne tout au long de ces difficiles années de thèse.

Je tiens tout d'abord à remercier Madame Yolaine Bourda, Chef du Département Informatique de Supélec de m'avoir accordé l'opportunité de réaliser cette thèse dans d'excellentes conditions.

Je tiens à exprimer ma profonde reconnaissance à Monsieur Christian JACQUEMIN, Professeur à LIMSI et à l'université de Paris11, d'avoir accepté la direction de ma thèse. Je le remercie pour sa disponibilité, ses conseils clairvoyants, son soutien et la confiance qu'il a bien voulu m'accorder.

Je tiens aussi à exprimer ma plus profonde gratitude à Madame Bich-Liên DOAN, Enseignante Chercheur à Supélec pour la proposition de ce sujet et pour l'intérêt et la disponibilité qu'elle a manifestés à l'égard de mes travaux de recherches ainsi que pour son soutien et sa patience. On a beaucoup travaillé et on a bien rigolé. Les conseils qu'elle m'a donnés et les discussions qu'on a pu avoir vont me manquer, ça c'est sûr. Qu'elle soit ici assurée de mon très grand respect et du plaisir que j'ai à travailler avec elle.

Je tiens également à remercier vivement Monsieur Mohand BOUGHANEM, Professeur à l'Université Paul Sabatier, et Monsieur Gilles FALQUET, Professeur à l'Université de Genève pour avoir accepté de m'accorder un temps précieux en tant que rapporteurs de cette thèse, ainsi que Madame Anne VILNAT Professeur au CNRS-LIMSI et à l'Université de Paris11 de me faire l'honneur de participer au jury.

Je souhaite aussi témoigner de ma sympathie et de ma gratitude aux membres du département informatique de Supélec avec qui j'ai eu le plaisir de travailler durant cette thèse, pour l'intérêt porté à mes travaux de recherche et leur accueil toujours chaleureux. Merci notamment à Nacéra BENNACER, Frédéric BOULANGER, Christophe JACQUET, Dominique MARCADET, Géraldine POLAILLON, Joanna TOMASIK, Safouan TAHA, Assia TOUIL, Guy VIDAL-NAQUET, Marc-Antoine WEISSER.

Je tiens à remercier en particulier Monsieur Philippe Porquet pour ces relectures pertinentes qui ont largement contribué à l'amélioration de la qualité de ce mémoire.

Mes remerciements vont également vers mes amis de toujours Cedric, Cécile, Laure, Lobna, Maran, Mamadou, Mbomi, Mohammed, Nadjet, Ounes, Rim, Vincent, Youssef qui ont toujours été là pour partager les soucis, les joies et les moments de détente. Merci d'avoir accepté que je sois moins disponible cette dernière année, aussi importante qu'elle ait été pour certains. Je salue également tous ceux qu'ils m'ont toujours été agréable de rencontrer au hasard d'un couloir ou d'un colloque.

Je ne voudrais pas oublier de remercier le personnel du CRI (Luc BATALIE, Noro BRISSAC, Claude BOCAGE, Stéphane BOZEC, Caroline CHARLES, Daniel CLAR, Eric COLLEAUX, Thibault LE MEUR, Sylvain VLACHOS pour leur bonne humeur et les services qu'ils m'avaient rendus.

Je salue aussi la disponibilité de Gilbert DAHAN et la gentillesse et l'efficacité de Madame Evelyne FAIVRE.

Mes remerciements vont également vers tous les membres de l'équipe SIGMI de l'université Paris Ouest Nanterre La Défense ex Paris10 (Michel BOUTILLIER, Claire HANEN, Gabriel HATCHIKIAN, Emmanuel HYON, Laurent MESNAGER, Laurent PIERRE, Saintenoy PATRICE, Ebrahimi REZA) pour leur convivialité et leur accueil chaleureux dans l'équipe pédagogique et la compréhension dont ils ont fait preuve en ce début d'année à l'emploi du temps chargé.

Mes plus affectueux remerciements vont évidemment à toute ma famille, et tout d'abord à mes parents Rachid et Addada qui m'ont toujours soutenu et encouragé dans tout ce que j'ai entrepris. Qu'ils soient ici en partie récompensés pour tout ce qu'ils m'ont donné. Une pensée particulière à mon frère Fouaz et mes deux sœurs Roza et Samia ainsi que leur maris Farid et Kader. J'adresse également un clin d'œil tout particulier à ma deuxième famille en France (Hamid, Saâdia, Chahine, Samia, Boubeker, Djaouida) qui ont toujours été d'un soutien inconditionnel, que ce soit par leur disponibilité et leur générosité.

Et bien sûr, toutes mes pensées vont à ma femme Taldja, que je remercie tendrement pour sa patience et tout l'amour qu'elle me porte. Rien de ce que j'ai entrepris d'important n'aurait pu se réaliser sans son soutien indéfectible, pour lequel lui suis infiniment reconnaissant. Enfin, à mon enfant Cyliane que j'adore beaucoup.

Résumé

Le Web est caractérisé par un volume d'information exponentiellement croissant ainsi que par l'hétérogénéité de ses ressources. Face au très grand nombre de réponses fournies par un moteur de recherche, il s'agit de fournir des réponses pertinentes parmi les premières réponses.

Nous nous sommes intéressés aux techniques de propagation de pertinence pour des corpus de documents hypertextes, et en particulier à l'analyse des liens afin d'exploiter l'information véhiculée par ses liens et par le voisinage des documents Web. Par définition, la propagation de pertinence consiste à propager des scores attribués à des pages à travers la structure du Web.

Cependant, la plupart des algorithmes de propagation de pertinence utilisent des paramètres fixes de propagation qui dépendent des requêtes exécutées et de la collection de documents utilisée. De plus, ces techniques ne distinguent pas entre les pages répondant totalement ou partiellement à la requête utilisateur et ne tiennent pas compte des différentes thématiques abordées dans les pages web. Pour améliorer ces techniques, nous avons proposé une nouvelle technique de propagation de pertinence en utilisant des paramètres calculés dynamiquement, indépendamment de la collection utilisée. En effet, nous avons proposé de modéliser une fonction de correspondance d'un système de recherche d'information en prenant en compte à la fois le contenu d'un document et le voisinage de ce document. Ce voisinage est calculé dynamiquement en pondérant les liens hypertextes reliant les documents en fonction du nombre de termes distincts de la requête contenus dans ces documents.

Pour traiter l'hétérogénéité des documents Web, Nous avons modélisé les ressources Web à différents niveaux de granularité (site, page, bloc) afin de prendre en compte les différents thèmes contenus dans un même document.

Nous avons proposé aussi une méthode de segmentation structurelle des pages Web reposant sur des critères visuels et de format des pages Web afin d'extraire des blocs thématiques qui seront utilisés pour améliorer les performances de la recherche d'information. Nous avons opté pour les algorithmes génétiques et l'analyse thématique afin de découper une page en plusieurs blocs thématiques. Ce choix est motivé par le fait qu'il existe plusieurs solutions de segmentation d'une page Web et que le problème qui se pose est comment choisir la meilleure solution parmi un ensemble de solutions ?

Afin de valider notre approche, nous avons expérimenté notre système sur deux collections de test WT10g et GOV issues de la compagne TREC. Les résultats obtenues montrent que notre modèle fournit de bons résultats par rapport aux algorithmes classiques reposant sur le contenu seul d'un document et ceux reposant sur l'analyse des liens (PageRank, HITS, propagation de pertinence). Nous avons constaté que les améliorations les plus importantes concernent le niveau bloc et que le gain en précision le plus significatif concerne la collection Gov.

Mots clés : Recherche d'information, propagation de pertinence, segmentation de pages Web, analyse des liens.

Abstract

The explosive growth of the web has led to surge of research activity in the area of information retrieval (IR) on the World Wide Web. Ranking has always been an important component of any information retrieval system (IRS). In the case of web search its importance becomes critical. Due to the size of the Web, it is imperative to have a ranking function that captures the user needs. The aim of the system is therefore to retrieve the set of documents which are most relevant to a query. To this end the Web offers a rich context of information which is expressed via the links. We are interested in the relevance propagation algorithms for hypertext document collections, specially, how to take advantage of link information and neighbourhood of documents in order to improve information retrieval.

In this thesis, we propose to model a new matching function for an information retrieval system using both the content and the neighbourhood of a web page. The neighbourhood of a web page is dynamically computed with the number of query terms that the pages are composed of. This function propagates scores from sources pages to destination pages in relation with the keywords of a query.

Indeed, we explore the use of web page topic segmentation algorithm based on visual criteria like the horizontal lines, colors, and content presentation of the page like headings <H1> <H6>, paragraph <P> and tables <TABLE> tags in order to separate possible segments of different topics and investigate how to take advantage of block-level evidence to improve retrieval performance in the web context.

We experienced our system over the two test collections WT10g and GOV. We conclude that our model provides better results in comparison with the baseline based-on text content only and those based-on link analysis (PageRank, HITS, relevance propagation).

Keywords : Information retrieval, Hypertexts Systems, Relevance Propagation, Web Page Segmentation, Link Analysis.

Table des matières

Liste des figures

Liste des tableaux

Chapitre 1

Introduction

1.1 Contexte

La Recherche d'Information (RI), née en 1950, s'attache à définir des modèles et des systèmes dont le but est de faciliter l'accès à un ensemble de documents sous forme électronique (corpus de documents), afin de permettre à des utilisateurs de retrouver les documents dont le contenu répond à leur besoin d'information. La RI est donc centrée sur la notion de pertinence qui est définie par le degré de corrélation entre la requête utilisateur et les réponses retrouvées. Les modèles de RI sont construits autour du triplet document, besoin d'information et fonction de correspondance. Ces modèles constituent encore aujourd'hui la base sur laquelle sont développés les systèmes de recherche d'information (SRI), dont les moteurs de recherche sur le Web. Ainsi, un SRI est un système qui indexe un corpus de document et qui évalue un ensemble de documents pertinents en réponse à une requête formulée par un utilisateur. Les systèmes de recherche d'information sont composés essentiellement de deux modules : un module d'indexation et un module d'interrogation. Le module d'indexation construit des abstractions des contenus de documents appelées index. Le module d'interrogation construit des abstractions des besoins d'information utilisateurs appelées requêtes et les compare à l'index grâce à une fonction de correspondance, laquelle permet de calculer une pertinence entre la requête et l'index. Cette fonction de correspondance est un composant très important dans tout SRI. Dans le cas de la recherche d'information sur le Web, son importance devient critique vu la taille du Web, qui atteint des milliards de documents. Il est donc impératif d'avoir de bonnes fonctions de correspondance afin de mieux répondre aux besoins d'information utilisateurs exprimés à travers les requêtes. L'architecture d'un SRI standard est illustrée dans la figure 1.1.

La qualité d'un système doit être mesurée en comparant les réponses du système avec les réponses idéales que l'utilisateur espère recevoir. Plus les réponses du système correspondent à celles que l'utilisateur espère, meilleur est le système. La démarche de validation en recherche d'information repose sur l'évaluation expérimentale des performances des modèles ou des systèmes proposés. Cette évaluation peut porter sur plusieurs critères : le temps de réponse, la pertinence, la qualité et la présentation des résultats, etc. Le critère le plus important est celui qui mesure la capacité du système à satisfaire le besoin d'information d'un utilisateur, c'est à dire la pertinence qui est une une notion complexe. Deux facteurs permettent d'évaluer ce critère. Le premier est le rappel, il mesure la capacité du système à sélectionner tous les documents pertinents. Le second est la précision, il mesure la capacité du système à ne sélectionner que les documents pertinents ou à rejeter tous les documents non pertinents. Les mesures de précision et de rappel sont très utilisées sur des corpus textuels lorsqu'on connaît l'ensemble des éléments du corpus analysé. Cependant, ces mesures sont difficilement applicables dans le cas d'un moteur de recherche car il est difficile d'avoir une idée précise de l'ensemble des documents visibles sur le Web.

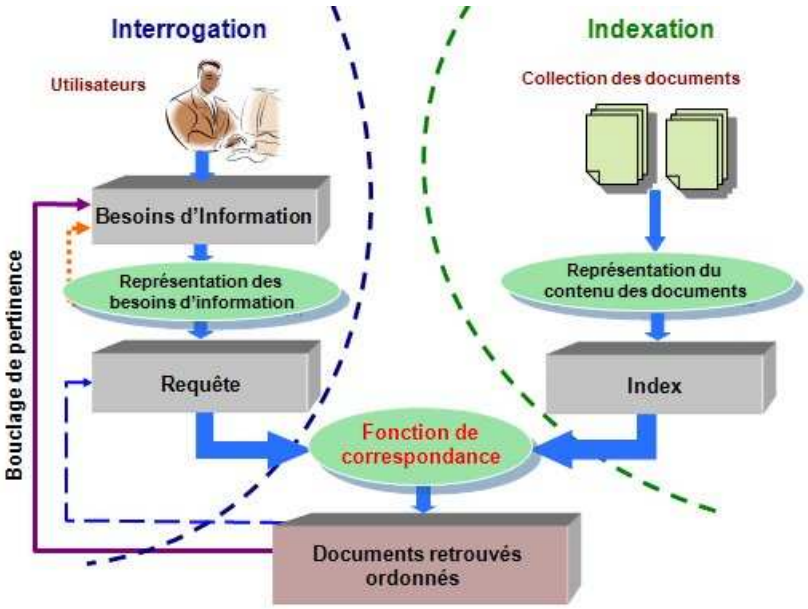


FIGURE 1.1: Architecture d'un système de recherche d'information (SRI)

Au fur et à mesure de l'évolution du domaine de la recherche d'information, d'autres méthodes standard de mesure de qualité telle que la précision moyenne MAP ("Mean Average Precision"), la précision à X documents retrouvés ont été mises au point afin de pouvoir comparer aisément des algorithmes différents de RI. Les mesures d'évaluation basées sur la notion de précision et de rappel comptent parmi les plus anciennes du domaine de la RI. Par definition, la précision est le rapport du nombre de documents pertinents retrouvés sur le nombre total de documents retrouvés ; alors que le rappel est le rapport du nombre de documents pertinents retrouvés sur le nombre total de documents pertinents. Considérons un exemple de besoin d'information et son ensemble P de documents pertinents. Soit $\mid P \mid$ le nombre de documents de cet ensemble. Supposons une stratégie donnée de recherche d'information qui traite ce besoin d'information et produit un ensemble de réponses R. soit $\mid R \mid$ le nombre de documents de cet ensemble. De plus, soit $\mid P_R \mid$ le nombre de documents de l'intersection des deux ensembles P et R. P_R est composé de documents pertinents au besoin d'information et retrouvés par la stratégie de recherche. La figure 6.1 illustre ces différents ensembles.

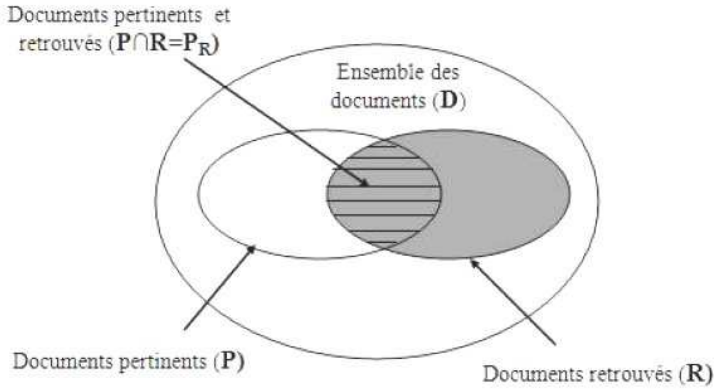


FIGURE 1.2: La précision et le rappel pour une requête donnée

Les mesures du rappel et de précision sont définies comme suit.

– Le rappel est la proportion de documents pertinents trouvés par rapport au nombre de documents pertinents.

$$Rappel = \frac{|P_R|}{|P|} \tag{1.1}$$

– La précision est la proportion de documents pertinents trouvés par rapport au nombre de documents trouvés.

$$Precision = \frac{|P_R|}{|R|} \tag{1.2}$$

1.2 Problématique

Avec le développement du Web, la quantité d'information indexée et accessible aux utilisateurs ne cesse de croître de manière exponentielle. Comme indication, une étude, menée conjointement par deux chercheurs des universités de Pise (Italie) et de l'Iowa (Etats-Unis) [GS05], donne une taille approximative de $11,5$ milliards de documents pour le Web "indexable" (ou "visible") en janvier 2005. L'étude tente d'indiquer également le taux de couverture réel de chaque moteur de recherche. Google, avec un index effectif de 8 milliards de pages, est en tête devant Yahoo ! $(6,6)$, Ask Jeeves $(5,3)$ et MSN $(5,1$ milliards). Toujours selon cette étude, et en tenant compte du taux de recouvrement entre les différents outils, $9,4$ milliards de pages (sur les $11,5$ au total) seraient "accessibles" en utilisant les moteurs de recherche. Bien sûr, ces informations ne tiennent pas compte du "Web invisible" dont la taille n'est pas mesurable. Ainsi, la RI doit faire face à de nouveaux défis d'accès à l'information, à savoir retrouver une information dans un espace diversifié et de taille considérable. Il est donc nécessaire d'avoir des outils performants pour une recherche efficace et effective. Le but d'un système de recherche d'information performant est donc d'arriver à afficher dans les dix à vingt premières réponses les documents répondant le mieux à la requête posée par l'utilisateur.

Dans la recherche d'information, obtenir une liste, la plus exhaustive possible, des sources répondant à une requête est nécessaire, mais insuffisant dès lors que le nombre de réponses dépasse la centaine. Il devient important de pouvoir discriminer, classer et évaluer tous ces résultats. L'utilisateur a besoin d'un ordre de lecture de toutes ces pages. Mais il peut aussi éprouver l'envie d'avoir une idée sur les différents thèmes abordés dans ces documents pour l'aider à mieux comprendre l'intégralité de l'information obtenue. Le principal outil d'aide proposé par les systèmes de recherche d'information est le classement des résultats, selon un indicateur souvent nommé « indice de pertinence ».

Dans la RI traditionnelle, la pertinence d'un document par rapport à la requête utilisateur réside dans son contenu seul. Par conséquent, les documents répondant à une requête utilisateur sont classés selon un degré de pertinence estimé pour chaque document et calculé en fonction de son contenu textuel. Ce degré de pertinence repose à la fois sur la fréquence d'apparition des termes de la requête dans la page et sur la localisation des termes (par exemple assigner des poids plus importants pour les termes qui apparaissent dans le titre, les metadonnées et au début de la page). Cet indicateur est utilisé systématiquement par les systèmes de recherche d'information traditionnels, de façon à classer le résultat d'une recherche par ordre d'intérêt décroissant. Les utilisateurs de ces systèmes ont pu vérifier, par expérience, du peu d'intérêt qu'a ce classement[GCH$^+$01a]. Il n'est pas rare de retrouver, en tête de liste, des pages Web qui ne sont pas du tout en adéquation avec la requête. En effet, le classement par pertinence a été altéré par le besoin par les auteurs de rendre leurs sites plus visibles. Par conséquent, les auteurs de sites se sont mis à étoffer le contenu de leurs documents à l'aide de techniques plus ou moins honnêtes, par exemple en surchargeant un document par des mots non visibles à l'utilisateur et indexés par les moteurs de recherche (par exemple ajout de mot-clés dans la balise <META>). Très souvent, au lieu de ne renvoyer que les documents pertinents, l'utilisateur se retrouvait alors face à des documents dont le contenu était à but commercial ou répondait à des critères de visibilité au lieu d'être réellement lié à sa requête.

Dans le cadre de corpus de documents hypertextes, l'information pertinente à une requête peut résider au-delà du contenu textuel du document. En effet, le web peut être représenté par un graphe dont les nœuds sont des documents et les arcs sont des liens qui relient ces documents. Nous distinguons deux types de liens : les liens informationnels qui apportent de l'information et les liens vides qui n'apportent pas d'information additionnelle aux documents (liens de publicité, liens de navigation). Or, les liens sont beaucoup plus nombreux que les documents dans le Web et donc représentent une source d'information très importante. Ainsi, la structure de graphe de liens peut être utilisée pour améliorer l'estimation de la pertinence accordée à chaque document. L'utilisation des liens pour améliorer la recherche d'information repose sur l'idée suivante : ***Les liens relient des documents sémantiquement proches et peuvent donc enrichir les documents en fournissant des informations additionnelles pour la recherche (description de contenu, importance en terme de popularité,etc)***.

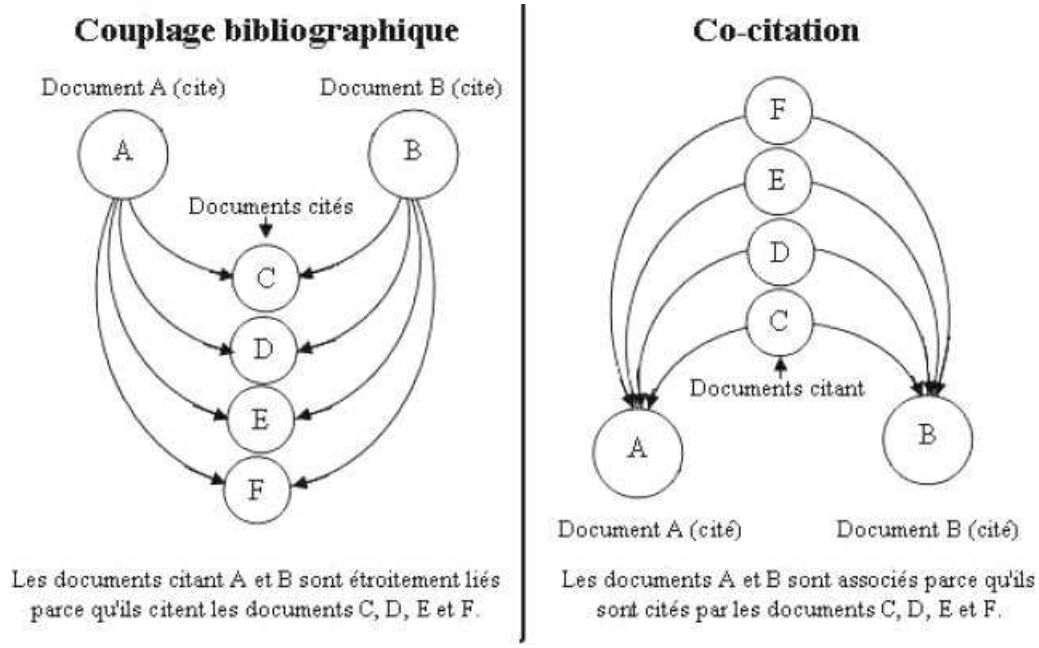


FIGURE 1.3: Couplage bibliographique Vs Co-citation

Depuis longtemps, les méthodes de citations, illustrées dans la figure 1.3, ont été utilisées en bibliométrie et ont pour objectif de créer à partir d'articles scientifiques d'un même domaine de recherche, et plus précisément de leurs références bibliographiques, des cartes relationnelles de documents ou d'auteurs qui reflètent à la fois les liens sociologiques et thématiques de ce domaine. Les relations bibliographiques entre les articles scientifiques sont des indices qui permettent de déduire la similarité entre eux. On distingue deux méthodes de citation des articles scientifiques utilisées en bibliométrie :

- La méthode des co-citations (Marshakova [Mar73] et Small [Sma73]) : la similarité entre deux articles est basée sur leur nombre de co-citations, c'est-à-dire le nombre de fois où ils sont cités ensemble par un autre article.
- Le couplage bibliographique (Kessler [Kes63]) : deux articles citant les mêmes travaux sont considérés comme très proches dans leur contenu et entretiennent donc des liens très forts, par le biais des travaux de référence communs.

Plusieurs bibliomètres (Rousseau [Rou97], Ingwersen et al. [Ing98], Aguillo [Agu99], Egghe [Egg00], Bjorneborn et al. [BI01] [Bjo01], Bjorneborn et al. [BI04], Thelwall et al. [TH03]) ont proposé des équivalences entre les concepts établis en bibliométrie et le graphe du Web. Plus particulièrement, Prime et al. [PBZ01], Larson [Lar96], Pitkow [Pit99], Brin et al. [BP98] et Kleinberg [Kle99] se sont intéressés à la transposition de la méthode des co-citations et de couplage bibliographique des documents pour caractériser le Web. Ces travaux de recherche mettent en évidence les limites théoriques et techniques de l'analogie entre la bibliométrie

et le Web. En effet, dans le réseau des publications scientifiques [Gar83], un lien de référence est considéré comme une citation et indique une relation importante entre la page qui contient la référence bibliographique et la page référencée. En effet, un article scientifique est évalué par un comité de lecture, d'où une évaluation des références bibliographiques que contient l'article. Par conséquent, s'il existe un lien entre deux articles scientifiques, cela veut dire que les deux articles sont étroitement liés et sémantiquement proches. Alors que sur le Web, un lien hypertexte ne matérialise qu'un lien de citation ou de référence. Une des limites de l'analogie entre la bibliométrie et le Web est de considérer tous les liens hypertextes comme des liens de citation ou de référence. Or, il existe des liens vides tels que les liens de publicité et les liens de navigation qui n'ont rien à voir avec le rapprochement sémantique des pages Web.

Les techniques d'analyse de liens ont été développées, premièrement, pour améliorer les performances de la recherche d'information sur le Web en calculant une valeur de pertinence d'un document en fonction non pas de son contenu seul mais également en fonction de son voisinage (documents reliés par des liens hypertextes), ainsi que de la structure globale du graphe. Deuxièmement, ces techniques nous permettent, dans une certaine mesure, et parmi d'autres techniques, d'atteindre et d'indexer des documents non visibles à l'utilisateur tels que les documents protégés, les bases de données, les documents multimedia (images, vidéos, etc). Enfin, elles peuvent servir à contrer le "***spamming de contenu***" (i.e : application de techniques malhonnêtes reposant sur la modification du document par l'ajout de mots-clés pour améliorer le score d'un document ou fausser la description du document). En effet, il est plus facile de modifier le contenu d'un document que de modifier le graphe de liens. C'est pour toutes ces raisons que les moteurs de recherche actuels tels que Google intègrent un module d'analyse de liens dans leur architecture.

Dans le contexte de collections de documents hypertextes homogènes traitant d'un seul thème (ex. collection d'articles scientifiques), le contenu informationnel de voisinage introduit par les liens tend à être de grande qualité et d'utilité. En effet, l'utilisation des méthodes d'analyse de liens appliquées à ces collections donne de meilleurs résultats par rapport aux techniques standards reposant sur le contenu seul [WSC+03]. Cependant, dans le cadre du web où les liens relient des documents hétérogènes de contenus différents, l'enrichissement des documents via les liens introduit du bruit et dégrade les performances de la recherche. En effet, les techniques d'analyse de liens montrent leurs limites dans les résultats de TREC sur le Web Track de 2000 et 2002 [SR00], [GCH+01b].

L'altération des performances de la recherche s'explique aussi par le fait qu'une page Web est souvent multi-thèmes et qu'elle est traitée comme entité indivisible quelle que soit sa longueur. De plus, il existe des parties d'une page web dites "***blocs***" qui n'ont rien à voir avec les thématiques de la page et qui faussent le calcul de pertinence de la page par rapport à un besoin utilisateur. Ces blocs correspondent aux barres de navigation, de publicité et d'offres commerciales et ne doivent pas intervenir dans le calcul de pertinence de la page. Ainsi, si nous considérons qu'une page Web n'est pas la plus petite unité d'information indivisible, la détection des blocs d'une page comme structures de contenus thématiques devient un facteur potentiel pour l'amélioration des performances de la recherche d'information sur le web. Par ailleurs, une étude montre que dans la perception humaine [Ber02], une page Web est perçue comme étant constituée de plusieurs objets différents plutôt que comme un seul objet. Par exemple, les utilisateurs distinguent les différents blocs d'une page web grâce aux indicateurs visuels contenus dans cette page tels que la couleur, les lignes horizontales et verticales et la représentation structurelle de la page (paragraphes, titres). La plupart des travaux de segmentation de pages Web utilisent ces critères afin de déterminer les frontières entre les différents blocs d'une page Web. Le problème qui se pose maintenant est comment choisir ces critères visuels sachant qu'il existe une multitude de critères visuels et que le choix des délimiteurs de blocs (ou segments) dépend de la conception de chaque page Web ? Enfin, des pratiques de "***spamming de liens***" ont été développées pour modifier la structure du graphe du Web afin de favoriser des documents par rapport à d'autres dans le classement des résultats. En effet, la course à la visibilité des documents ou sites Web et la concurrence entre les différents acteurs du Web ont poussé les

auteurs de sites commerciaux et publicitaires à faire plus d'efforts pour rendre leur site visible dans les premières pages des résultats retournés par un moteur de recherche, même si cela s'avère coûteux et laborieux : négociation avec les auteurs d'autres sites sur l'ajout de liens et de mots-clés, création de sites virtuels pour ajouter des liens, achat de liens et de mots-clés, conception de générateurs automatiques de liens artificiels à partir d'autres documents tels que les forums et les blogs. Il est donc nécessaire de penser à d'autres techniques pour résoudre ou limiter ce nouveau problème de spamming des liens.

1.3 Objectifs et contributions de la thèse

Dans ce travail de thèse, nous proposons une nouvelle manière de calculer une valeur de pertinence pour un document par rapport à un besoin utilisateur. Notre technique permettra de résoudre les problèmes de spamming sur le Web et de contrer les limites des techniques d'analyse de liens et celle du contenu. En effet, le fait d'utiliser la structure du graphe de liens pour enrichir la pertinence des différents documents répondant à un besoin utilisateur est certainement bénéfique pour la recherche d'information. Nous pouvons utiliser d'une manière efficace une masse d'information très importante en provenance des liens afin de choisir les informations pertinentes à un besoin utilisateur. Notre approche est sensible à la requête. Elle dépend des termes de la requête dans le calcul de pertinence des documents par rapport à un besoin utilisateur. Nous donnons plus d'importance aux documents qui satisfont totalement la requête, ainsi que le voisinage d'un document par rapport à la requête. Nous nous sommes intéressé aux larges collections de documents et l'objectif attendu n'est pas de retrouver toutes les réponses qui satisfont un besoin utilisateur mais plutôt d'améliorer la précision de la recherche pour les N documents retrouvés en tête du classement. En effet, lors de la recherche, l'utilisateur ne consulte que les documents figurant dans les premières pages d'un moteur de recherche et dans plusieurs cas, il ne dépasse même pas la première page. D'où, l'intérêt de renvoyer les documents les plus pertinents dans la première page. Le deuxième point abordé dans la thèse est la granularité de l'information à renvoyer à l'utilisateur. De ce fait, nous nous sommes notamment intéressé à la segmentation des pages Web en utilisant des critères visuels contenus dans ces pages afin de les découper en différents blocs thématiques. Nous nous sommes intéressé à la tâche dite d'analyse thématique, qui vise l'étude de la structure des pages web selon des critères relatifs à la répartition de leur contenu informationnel. Ces travaux concernent donc le problème de la segmentation automatique des pages web ainsi que la représentation des thèmes des segments textuels que nous identifions. Enfin, nous proposons une architecture à trois niveaux, qui nous permet de traiter la pertinence à différents niveau d'abstraction (site, page et blocs).

1.4 Plan de la thèse

Le chapitre 2 présente les méthodes d'analyse de liens les plus connues, comme le PageRank de Brin et al. [BP98] et HITS de Kleinberg [Kle99]. Ces méthodes sont présentées, car elles sont à l'origine de beaucoup de travaux de recherche sur l'analyse des liens construits pour améliorer la précision dans la recherche d'information. Nous avons classé les différentes méthodes d'analyse de liens étudiées dans l'etat de l'art en deux grandes familles : la propagation de popularité et la propagation de pertinence. En effet, la propagation de popularité repose sur une distribution de probabilité sur les pages Web. Le but d'une telle approche est d'attribuer une valeur d'importance pour chaque document Web en les traitant équitablement. Les algorithmes de cette famille sont soit dépendants de la requête, soit indépendants de la requête. Tandis que la propagation de pertinence consiste à propager des scores de pertinence à travers la structure des liens. Avec ces dernières techniques, les documents ne sont pas traités équitablement. Dans ce chapitre, nous avons aussi présenté les travaux de recherche d'analyse de liens au niveau de granularité d'information plus petite que la page qui est le bloc d'information thématique. Ces méthodes calculent une certaine pertinence au niveau bloc. Des approches

mettant en oeuvre la segmentation des pages Web en plusieurs blocs sont également présentées. Nous ne nous sommes intéressé qu'aux méthodes de segmentation quantitatives dites numériques. En effet, les méthodes linguistiques de segmentation de pages Web ne peuvent pas être appliquées à de grands volumes de documents. Ces dernières nécessitent des ressources considérables afin de découper un texte en plusieurs paragraphes de thématiques différentes. Nous distinguons deux classes de méthodes de segmentation numériques : la segmentation linéaire qui découpe une page Web en blocs thématiques adjacents et la segmentation hiérarchique qui découpe une page Web de manière Top-Down ou Button-up en utilisant la structure hiérarchique de l'arbre DOM (Document Object Model) de la page. De plus, selon le mode de découpage utilisé afin d'extraire les segments de thématiques différentes, nous avons classé ces méthodes de segmentation en deux catégories : logique et physique. Les méthodes de segmentation logique consistent à determiner la probabilité d'une page par rapport une thématique donnée calculée en utilisant les distance entre les termes des blocs et les termes constituant le thème. Tandis que les méthodes de segmentation physique représentent le découpage physique de pages Web en blocs thématiques.

Le chapitre 3 est consacré à la description de notre modèle. Nous proposons une première contribution qui est la modélisation d'une fonction de correspondance qui tient compte à la fois du contenu d'une page et du voisinage de cette page. Ce voisinage est calculé dynamiquement en pondérant les liens hypertextes reliant les pages en fonction des termes de la requête contenus dans ces pages. Nous présentons ensuite un modèle d'architecture en trois couches (bloc, page et site), représentant chacune un niveau d'abstraction sur lequel va s'appliquer notre fonction de correspondance. La construction des niveaux page et site est faite à partir des graphe des pages Web, alors que la constructure du niveau bloc est réalisée à partir d'un algorithme de segmentation. La troisième partie de notre contribution repose sur notre algorithme de segmentation de pages Web en blocs thématiques reposant sur les critères visuels (i.e. lignes horizontales <HR>) et les critères de représentation du contenu des pages HTML (titres <H1>, paragraphes <P>, <TABLE>). Cet algorithme consiste à évaluer plusieurs solutions de segmentation et de choisir parmi elles la meilleure solution. Nous avons fait une analogie entre le problème de segmentation des pages Web et les algorithmes génétiques et nous avons proposé une fonction d'évaluation d'une solution de segmentation reposant sur la cohérence du contenu textuel à l'intérieur du bloc et des distances entre les blocs adjacents d'une segmentation.

Le chapitre 4 montre les différentes expérimentations effectuées sur deux collections issues de la conférence TREC qui sont WT10g et Gov. Nous concluons que notre fonction réalise de bons résultats par rapport à l'algorithme classique reposant sur le contenu seul de la page et ceux reposant sur la propagation de popularité ou la propagation de pertinence classiques.

Enfin, le chapitre 5 conclut ce travail en mettant en avant notre contribution, et en présentant les différentes perspectives qui en découlent.

Chapitre 2

Modèles d'analyse de liens en recherche d'information

2.1 Introduction

Afin de mettre l'apport proposé dans ce mémoire dans la perspective des travaux publiés sur le même sujet, nous consacrons ce chapitre à une présentation des approches les plus similaires à la nôtre. Le nombre très important de publications relatives à l'utilisation des liens en recherche d'information rend impossible une présentation exhaustive de toutes ces méthodes. Nous insistons plus particulièrement sur les travaux les plus connus qui portent sur la propagation de valeurs (pertinence ou popularité) à travers le graphe du Web. Nous portons une attention particulière aux approches que nous utilisons dans nos experimentations afin d'évaluer notre système par rapport à ces approches, en insistant sur les avantages et les limites de chaque approche. Nous portons aussi une une attention particulière à la recherche thématique sur le Web. En conclusion, nous exposons une étude comparative de toutes ces approches.

Les modèles d'analyse de liens que nous présentons dans les sections qui suivent sont inspirés des travaux de recherche issus de la bibliométrie. On distingue deux méthodes de citation des articles scientifiques utilisées en bibliométrie : la co-citations (Marshakova [Mar73] et Small [Sma73]) et le couplage bibliographique (Kessler [Kes63]). Nous avons classé ces modèles en trois grandes familles selon leur mode de fonctionnement 2.1, à savoir : la propagation de popularité, la propagation de pertinence et la propagation d'information. Il existe plusieurs travaux de recherche qui portent sur l'analyse des graphes du Web afin d'améliorer les performances de la recherche d'information. Par definition, le graphe du Web est constitué de plusieurs nœuds qui représentent les documents et de liens qui relient ces nœuds. Nous distinguons deux types de liens dans le Web : les liens hypertextes et les liens de structure physique des sites Web (chemin d'accès à un document ou URLs). Il existe aussi plusieurs types de document : pages Web, blocs, chemins de lecture, groupes de pages Web (ex. un site). Dans les algorithmes d'analyse de liens, la propagation de popularité, de pertinence ou d'information à travers les liens du graphe permettent d'accorder une certaine importance aux nœuds du graphe qui jouent un rôle particulier (portail, page d'accueil, document de référence, etc.), dans le but d'enrichir sémantiquement le contenu de ces nœuds et d'augmenter leur pertinence par rapport à un besoin utilisateur. La propagation d'information porte sur le contenu des nœuds citant le nœud courant (par exemple associer le texte ancre des liens, les mots clés et les titres de la page source du lien à la page destinataire du lien). Tandis que la propagation de popularité et la propagation de pertinence consistent à assigner une nouvelle valeur de qualité au nœud courant en fonction des valeurs de qualité des nœuds citant ou cités par ce nœud courant. Nous distinguons deux types de valeurs de qualité : la popularité et l'indice de pertinence. Lors du calcul de la popularité d'un

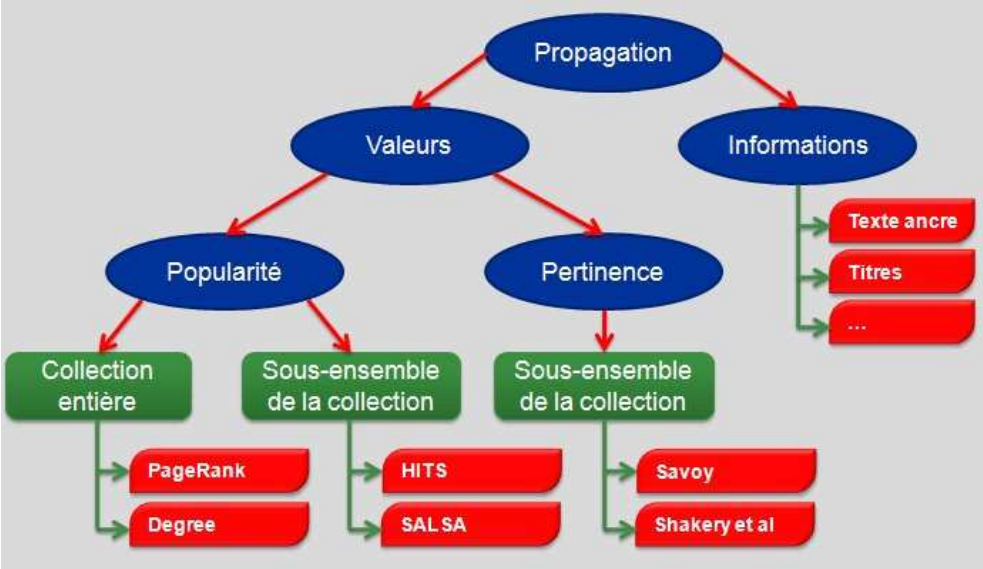


FIGURE 2.1: Types de propagation en recherche d'information

nœud, tous les nœuds du graphe sont traités équitablement, c'est-à-dire qu'ils ont les mêmes valeurs de qualité avant la propagation, ces valeurs sont calculées selon une distribution de probabilité. Au contraire, l'indice de pertinence est calculé en fonction de la requête. Dans ce cas de figure, les nœuds ont des valeurs de qualité différentes avant la propagation qui correspond à l'indice de pertinence du document par rapport à la requête utilisateur. Nous ne nous intéressons qu'aux modèles de propagation de valeurs (pertinence ou popularité). Ce choix est motivé par la similitude de notre approche de recherche à celles-ci. De ce fait, nous présentons dans ce qui suit les origines et les différentes catégories de chaque modèle de propagation de valeurs utilisées dans les travaux de recherche antérieurs.

2.2 Propagation de popularité

La propagation de popularité est une approche dérivée des méthodes d'analyse de citations et de co-citations utilisées en bibliométrie (Kessler [Kes63], White et al. [WM89]) qui consiste à privilégier les documents populaires qui jouent un rôle particulier dans le graphe de liens. Typiquement, il s'agit de la notion de popularité : ***"une page référencée par un grand nombre de pages est une bonne page"***. Différentes études ont suggéré de tenir compte de la popularité des documents afin d'améliorer les performance de la recherche d'information. Le PageRank [BP98] de Google et le HITS [Kle99] de Kleinberg sont deux algorithmes fondamentaux qui utilisent les liens hypertextes pour classer les résultats d'une requête. Un certain nombre d'extension de ces deux algorithmes ont été proposés, comme Lempel [LM00], Haveliwala [Hav02], Kamvar et al. [KHMG03], Cai et al. [CYWM04], Jiang et al. [JXS+04] et Jeh et all [JW03]. Généralement, ces algorithmes fonctionnent en deux temps. Dans une première étape, un moteur de recherche classique retourne une liste de documents répondant à la requête posée, en fonction des termes de la requête et des termes d'indexation des documents. Dans une seconde étape, ces systèmes tiennent compte des liens hypertextes pour classer ces documents. Les scores accordés aux documents avant la propagation sont identiques. L'utilisation d'un algorithme d'analyse de liens permet en sorte de calculer une valeur de la popularité de chaque document répondant à la requête utilisateur. Nous distinguons deux catégories d'algorithmes d'analyse de liens : la première catégorie regroupe tous les algorithmes qui calculent une valeur de popularité unique à l'étape d'indexation, c'est le cas du PageRank [BP98] qui calcule une valeur de popularité pour l'ensemble des pages Web en appliquant l'algorithme

de PageRank sur le graphe du Web en entier. La deuxième catégorie regroupe les algorithmes d'analyse des liens appliqués à un sous-ensemble de pages Web répondant totalement ou partiellement à la requête utilisateur. Nous avons choisit le PageRank [BP98] et HITS [Kle99] comme deux algorithmes représentatifs de ces deux catégories et nous les avons utilisés dans nos expérimentations afin de les comparer par rapport à notre approche et celle de la propagation de pertinence que nous étudierons ultérieurement dans la section 2.3.

2.2.1 Propagation de popularité sur l'ensemble d'une collection

Dans les systèmes de propagation de popularité appliquée à l'ensemble des documents de la collection, le contenu des documents et la structure des liens ont été utilisés séparément (voir la figure 2.2). C'est le cas des approches comme Hawking [Haw00], Craswell et al. [CHWW03] et Craswell et al. [CH04] qui utilisent le modèle vectoriel [SWY75] pour calculer un degré de pertinence reposant sur le contenu du document et les liens hypertextes pour calculer un indice de popularité indépendant de la requête (exemple PageRank [BP98]). Ensuite, ces deux scores sont combinés pour classer les documents retrouvés. Ces méthodes sont avérées en pratique moins efficaces par rapport aux méthodes utilisant uniquement le contenu seul des documents.

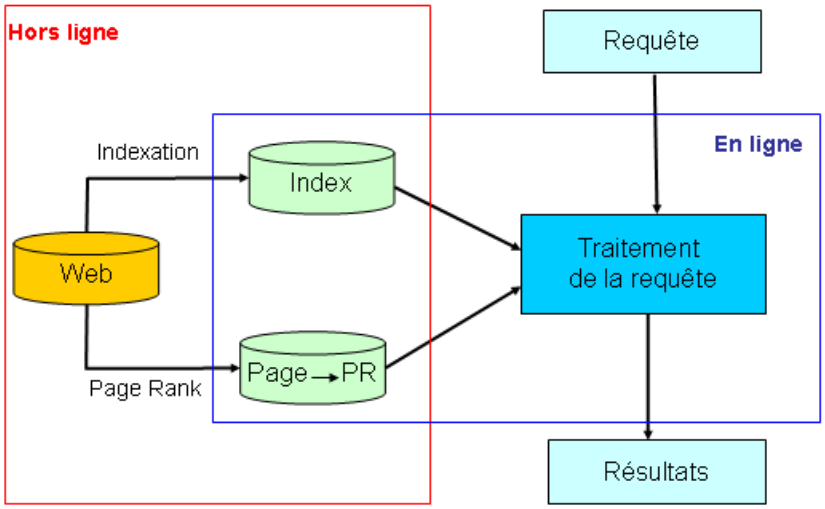


FIGURE 2.2: Architecture d'un SRI utilisant le PageRank

Dans la suite, nous allons vous présenter deux algorithmes existants les plus connus qui reposent sur la notion de popularité des pages indépendamment de la requête.

2.2.1.1 InDegree

Degree, en incluant InDegree et OutDegree, était parmi les premières métriques d'analyse des réseaux sociaux au cours des années 1930 dans le domaine de la bibliométrie et la sientométrie. Ils ont été utilisés avant l'arrivé des ordinateurs parce qu'ils étaient faciles à calculer. Il suffit de compter le nombre de références bibliographique citant ou cités par les articles scientifiques. InDegree est un algorithme simpliste qui utilise la popularité d'une page comme un facteur de classement des pages retournées par un système de recherche d'information. La popularité d'une page est calculée à partir du nombre de pages la citant, semblable à la technique de citation des documents (Marshakova [Mar73] et Small [Sma73]) dans la bibliométrie. Aux premiers jours du Web, cet algorithme a été largement utilisé par les moteurs de recherche. Cependant, l'algorithme InDegree n'est pas efficace pour deux raisons :

– il prend en considération tous les liens partant de n'importe quelle page, mais pas que ceux qui sont pertinents à la requête.
– l'algorithme peut être facilement influencé en ajoutant par exemple de milliers de liens partant de tous les côtés du Web, en augmentant ainsi artificiellement la popularité de la page.

La popularité $InDegree(P)$ d'une page P est calculée par la formule suivante :

$$InDegree(P) = \frac{|E(P)|}{|V|} \tag{2.1}$$

Où $E(P)$ est l'ensemble de pages citant la page P et $|E(P)|$ le nombre de pages dans l'ensemble $E(P)$. V est l'ensemble de pages constituant l'index du système et $|V|$ le nombre d'éléments de cet ensemble.

2.2.1.2 PageRank

Quelques moteurs de recherche, dont le plus connu est Google, ont pris le pari d'utiliser un autre mode de classement des résultats. Les pages Web sont ordonnées selon leur notoriété(popularité). Ce principe est directement inspiré de recherches antérieures en bibliométrie et principalement des travaux de Price [dSP63] et Garfield [Gar72] [Gar83] sur la pratique de la citation entre les articles scientifiques. Cette théorie veut qu'un article scientifique fréquemment cité par d'autres scientifiques fait partie des meilleurs articles. La citation serait comme une mesure du pouvoir d'utilité d'un article et par là même une certaine marque de qualité. Appliquant cette théorie à l'espace hypertextuel du Web, une page qui est la cible d'un très grand nombre de liens est probablement non seulement une page validée (page parcourue par un grand nombre de lecteurs, qui ont jugé bon de la citer en référence) mais aussi une page détenant un contenu utile à un grand nombre d'utilisateurs.

Le PageRank, introduit par Page et al. [BP98] en 1998, est la méthode de classement qui a fait la spécificité du moteur de recherche Google. Il s'agit en fait d'une adaptation au Web de diverses méthodes de citations introduites dans la bibliométrie. L'approche du PageRank repose sur la notion de propagation de popularité. Le principe consiste à évaluer l'importance d'une page en fonction de chacune des pages pointant vers elle. La propagation met en avant les pages qui jouent un rôle particulier dans le graphe des liens, avec l'hypothèse suivante : ***"une page est importante quand elle est beaucoup citée ou citée par une page très importante"***. La mesure de PageRank (PR) proposée par Page et al. [BP98] est une distribution de probabilité sur les pages. Elle mesure en effet la probabilité PR, pour un utilisateur navigant au hasard, d'atteindre une page donnée. Elle repose sur un concept très simple : un lien émis par une page A vers une page B est assimilé à un vote de A pour B. Plus une page reçoit de votes, plus cette page est considérée comme importante. Le PageRank se calcule de la façon suivante :

Soient T_1, T_2, ..., T_n : n pages citant une page A. Notons $PR(Tk)$ le PageRank de la page T_k, $S(T_k)$ le nombre de liens sortants de la page T_k, et d un facteur compris entre 0 et 1, fixé en général à $0,85$. Ce facteur d représente la probabilité de suivre effectivement les liens pour atteindre la page A, tandis que (1-d) représente la probabilité d'atteindre la page A sans suivre de liens. Le PageRank de la page A se calcule à partir du PageRank de toutes les pages T_k de la manière suivante :

$$PR(A) = (1-d) + d\sum_{i=1}^{n} \frac{PR(T_i)}{S(T_i)} \tag{2.2}$$

Intuitivement, la formule 2.2 signifie que le PR d'une page A dépend à la fois de la qualité des pages qui la citent et du nombre de ces pages. Par exemple, si nous considérons la page d'accueil de Yahoo !, ayant un PR élevé, les pages citées par elle seront jugées de bonne qualité.

La détermination des valeurs du PageRank des pages Web n'est pas un calcul simple. Selon Google [1], il existe 1 trillion de liens hypertextes dans son index recensés en Juillet 2008. En fait, ce calcul est appelé le plus grand calcul matriciel du monde [Mol02]. Les méthodes exactes ne peuvent pas manipuler la taille phénoménale de la matrice. Donc, des variantes des méthodes puissantes semblent être le seul choix dans la pratique Chien et al. [CDKS02], Kamvar et al. [KHG03], Ng et al. [NZJ01a] [NZJ01b], Langville [LM02] . Le temps requis par Google pour calculer le vecteur de PageRank est de l'ordre de plusieurs semaines. Les caractéristiques de l'algorithme du PageRank sont :

- La valeur du PR d'une page A, est calculée par un algorithme itératif. Initialement, toutes les pages sont équiprobables, leur valeur de PR est alors égale à $1/n$, n étant le nombre de documents de la collection. Un cycle d'itération est nécessaire pour propager les probabilités sur les pages. L'algorithme s'arrête théoriquement lorsqu'une nouvelle itération ne produit plus de modifications des valeurs de PR des pages. En pratique la convergence est obtenue au bout de plusieurs dizaines d'itérations (tout dépend du nombre de pages considérées).
- Le calcul du PageRank se fait off-line (lors de la phase d'indexation) indépendamment de la requête. Lors de l'interrogation du système, seulement une consultation rapide dans l'index est nécessaire pour déterminer les scores de pertinence des pages en réponse à une requête utilisateur. la pertinence d'une page par rapport à la requête est calculée en fonction de la valeur de PR et d'une autre mesure dépendant des termes de la requête (ex. tfidf [SWY75]).

Bien évidemment, l'indice de PageRank a des failles, tout comme la citation en bibliométrie. Parmi les failles de l'indice de PageRank, on peut citer l'auto-citation, l'avantage cumulé, le spamming des liens et le sens des thèmes.

- L'auto-citation (Le fait qu'un auteur cite ses propres travaux), dans le monde du Web, correspond aux liens pointant sur une page alors qu'ils proviennent d'une page du même site. Ces liens sont probablement des liens de navigation du site lui-même, liens essentiellement utiles au parcours du site. Il paraît juste de ne pas les prendre en considération lors du calcul.
- Le phénomène d'avantage cumulé, qui symbolise le fait qu'on s'intéresse qu'aux grand sites, se vérifie là encore. Plus une page ou un site sera pointé par un grand nombre de liens, plus la probabilité d'y accéder sera grande, plus forte sera la probabilité qu'elle soit de nouveau la cible de prochains liens. Le taux de citations reçues par une page Web ne présume donc pas forcément de la qualité de son contenu, mais tout au moins de sa popularité, voir seulement de sa visibilité dans le Web.
- Une autre faiblesse de l'indice de PageRank est le problème de sens des thèmes. Comme le calcul de PageRank est indépendant de la requête, les documents auront le même classement. Il suffit qu'un document important contient les termes de la requête pour figurer au top du classement. Beaucoup de travaux, des idées et des heuristiques nécessitent d'être appliquées par des ingénieurs de Google pour déterminer les poids de pertinence des documents par rapport à la requête, sinon, la liste renvoyée en appliquant un PageRank brut pourrait être inutile à l'utilisateur. Bharat et al [BM02] ont brièvement mentionné cet inconvénient de PageRank dans leurs travaux. Puisque le PageRank est indépendant de la requête, il ne peut pas distinguer entre les documents importants en général et les documents importants pour un thème particulier de la requête. Pour résoudre ce problème, Haveliwala [Hav02] a proposé une solution pour biaiser le calcul du PageRank en donnant plus d'importance aux pages bénéficiant de liens entrants en provenance de sites dont la thématique est connue. Le but était de calculer autant de vecteur PageRank que le nombre de thèmes existants dans le Web et d'utiliser seulement le vecteur de PageRank associé à la thématique de la requête lors de l'interrogation du système.
- L'application des techniques de spamming de liens peut fausser les calculs. En effet, Chien et al [CDKS02] ont prouvé que si le propriétaire d'un document réussit à ajouter des liens qui pointent son document dans des documents importants en achetant ces liens, le PageRank de ce document est garanti d'augmenter.

1. The Officiel Google Blog (25/07/2008) : **"We knew the Web was big..."**// http ://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html

De ce fait, le PageRank de ce document ne reflète pas son importance. Néanmoins, il existe quelques articles qui décrivent des manières pour influencer le calcul de PageRank et identifier les tentatives de spamming Bianchini et al. [BGS05], Tsoi et al. [TMS+03]. Cependant, il est difficile de distinguer entre les liens informationnels qui apportent de l'information et les liens vides.

Comme notre travail s'inscrit dans le cadre des approches fondées sur le calcul des scores de pertinence à plusieurs niveaux de granularité d'information (blocs thématiques, pages et sites Web), nous proposons dans les paragraphes suivants quelques extensions de l'algorithme du PageRank utilisées dans la littérature. L'idée de ces algorithmes est de calculer des PageRanks personnalisés. Nous distinguons deux solutions radicalement différentes (Haveliwala [HKJ03]) :

- Le PageRank modulaire
- Le PageRank calculé par blocs ("BlockRank")

Chacune de ces deux approches calcule une certaine approximation de PR. Cependant, elles diffèrent considérablement dans leurs conditions de calcul et dans la granularité de la personnalisation réalisée. Dans ce qui suit, nous décrivons chacune d'elles.

2.2.1.2.1 Le PageRank modulaire : L'approche du PageRank modulaire proposé par Jeh et al. [JW03] part du principe qu'il est possible de calculer une bonne approximation des PR de toutes les pages en découpant le Web à partir d'un ensemble de pages de départ appelé ensemble de pages pivots (EPP). L'ensemble des pivots peut être composé de pages dont le PR est le plus élevé (les pages très importantes), de pages mentionnées dans l'ODP (Open Directory Project) de Yahoo ! ou les pages importantes pour une entreprise particulière (comme Microsoft). A partir de chaque page pivot, on construit des zones de pages. Une zone est composée de toutes les pages qu'on peut atteindre à partir de la page pivot. L'ensemble des zones déterminées à partir de l'ensemble des pivots doit être un sous ensemble du Web plus ou moins complet (i.e. l'union de ces zones couvre le Web entier). Les meilleurs résultats sont obtenus lorsque l'ensemble des pivots est composé de pages dont le PR est élevé. Dans ce cas, on obtient des résultats proche du PageRank global. Cependant, il est difficile de trouver un ensemble de pivots qui permet d'obtenir un recouvrement quasi complet du Web. En effet, il existe plusieurs pages isolées qui n'ont pas de liens entrants, donc inaccessibles à partir d'une des pages pivots de l'ensemble EPP.

2.2.1.2.2 Le BlockRank : Cet algorithme part d'un constat expérimental : les liens entre les pages Web ne sont pas répartis uniformément. En effet, il existe des groupes de pages fortement inter-connectées, comme ceux constituant un domaine, un site Web ou un répertoire. Ces groupe de pages sont ensuite reliés entre eux par un nombre faible de liens (liens qui relient entre des pages de deux groupes différents). Des expériences ont montré qu'environ 79% des liens se trouvent dans le même site. De même, environ 84% des liens se trouvent dans le même domaine [JXS+04]. Le Web possède une réelle structure en blocs. On distingue nettement des blocs imbriqués correspondants aux domaines, sous domaines et sous répertoires. Les blocs sont bien plus petits que le Web en entier. Dans l'architecture proposée par [JXS+04], on distingue quatre types de bloc : la page elle-même, le répertoire, le site et le domaine (voir la figure 2.4).

L'idée de l'algorithme du BlockRank consiste dans une première étape à calculer des PageRank locaux par blocs. Avec un nombre réduit de page (prendre en compte que les pages constituant un bloc), l'algorithme PageRank converge rapidement. La deuxième étape, pour obtenir une excellente approximation du PageRank global, consiste à calculer l'importance du bloc en se basant sur une matrice de blocs, et non pas le Web entier. On note cette valeur BlockRank[KHMG03]. Le calcul du BlockRank est défini comme suit :

- Découper le Web en blocs selon le type de bloc (domaine, site ou répertoire)
- Calculer le PageRank local de chaque page dans un bloc
- Estimer l'importance relative de chaque bloc (notée BlockRank)

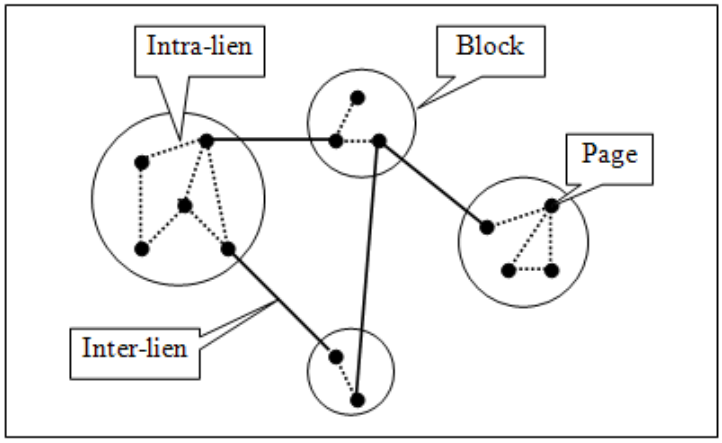


FIGURE 2.3: Les différents types de liens dans dans le modèle BlocRank

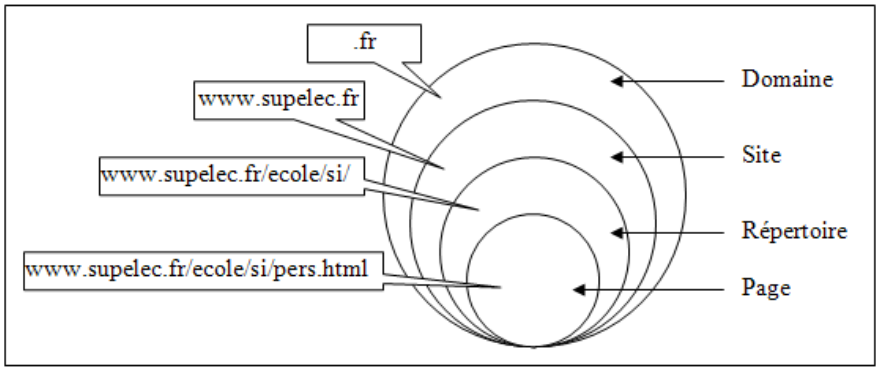


FIGURE 2.4: Les différents blocs dans dans le modèle BlocRank

– Pondérer le PageRank local de chaque page par le BlockRank du bloc de la page et les regrouper pour former une estimation du PageRank global.

Cet algorithme représente une avancée considérable : il diminue le temps de calcul des PageRanks de manière considérable (jusqu'à 20 fois moins). Il permet aussi de déterminer une bonne approximation du PR effectif pour toute nouvelle page entrante dans l'index. Les avantages principaux de l'algorithme du BlockRank sont les suivants :

– L'accélération du calcul vient principalement du fait que le calcul du PageRank local (par bloc) peut être fait en mémoire, et non pas avec de nombreux accès à un disque dur.
– Le calcul du PageRank local converge très rapidement pour la majorité des blocs. Dans l'algorithme classique, le calcul est ralenti à cause du nombre de pages à considérer.
– Le calcul du PageRank local peut être largement parallélisé. De plus, il peut même être prévu lors de l'indexation de chaque bloc (le calcul peut démarrer dès que tout un sous-domaine a été totalement indexé).
– Dans certains cas, il est nécessaire de recalculer le PageRank global alors que peu de changements ont eu lieu. Avec l'algorithme du BlockRank, il suffit de recalculer le PageRank local du bloc qui a changé. Par exemple, on peut recalculer le PageRank global en tenant compte des mises à jour fréquentes d'un site, sans avoir à recalculer le PageRank local de tous les autres sites.

2.2.2 Propagation de popularité sur les résultats d'une requête

Dans la premiere catégorie de propagation de popularité exposée dans la section 2.2, nous avons vu que l'indice de popularité de chaque document de la collection est calculé indépendamment de la requête utilisateur. En effet, le résultat de ce calcul, réalisé off-line, est un vecteur d'importance des documents stocké dans l'index de la collection. Dans cette deuxième catégorie, la propagation de popularité est appliquée à sous-ensemble de la collection associé à la requête utilisateur et composé de documents contenant les termes de la requête. En effet, les documents sont retrouvés selon l'existence ou non des termes de la requête utilisateur dans ces documents, puis des techniques d'analyse de liens qui s'inspirent des études de graphes sont appliquées pour classer ces documents. Kleinberg [Kle99] et Lempel et al. [LM00] sont deux algorithmes représentatifs de cette catégorie que nous allons discuter dans ce qui suit. Nous détaillons plus en détail l'algorithme de HITS [Kle99] que nous allons utiliser dans nos experimentations afin de le comparer par rapport à notre approches reposant sur la propagation dynamique de pertinence et d'autres techniques d'analyse de liens reposant sur la propagation de popularité indépendamment de la requête utilisateur [BP98] et la propagation statique de pertinence que nous allons voir dans la section 2.3.

2.2.2.1 HITS

Un indicateur peut être élaboré à partir du phénomène de référencement entre pages Web : le ***pouvoir rayonnant***. Plus une page Web contient de liens vers d'autres pages Web importantes plus son pouvoir rayonnant est important. Le projet Clever d'IBM (Kleinberg [Kle99]) a même perfectionné cette mesure en donnant un poids plus fort aux liens pointant des pages de très forte notoriété. Plus une page fait référence à de nombreuses pages fortement citées, plus son pouvoir rayonnant s'amplifie. Les pages Web ayant un fort pouvoir rayonnant sont nommées pages pivot au sein du projet Clever. Les chercheurs impliqués dans ce projet ont très rapidement constaté que le système de classement qu'offrent les systèmes de recherche d'information, fondé sur le calcul des occurrences / localisations des termes de la requête, n'était pas assez significatif. Ils ont donc cherché à améliorer la qualité de ce classement en appliquant la théorie de la citation et plus particulièrement les travaux sur la mesure du facteur d'impact mis au point par Garfield [Gar72]. L'objectif est de détecter puis de classer les pages appartenant aux deux catégories : pages populaires (appelées ***"pages autorités"***) et pages

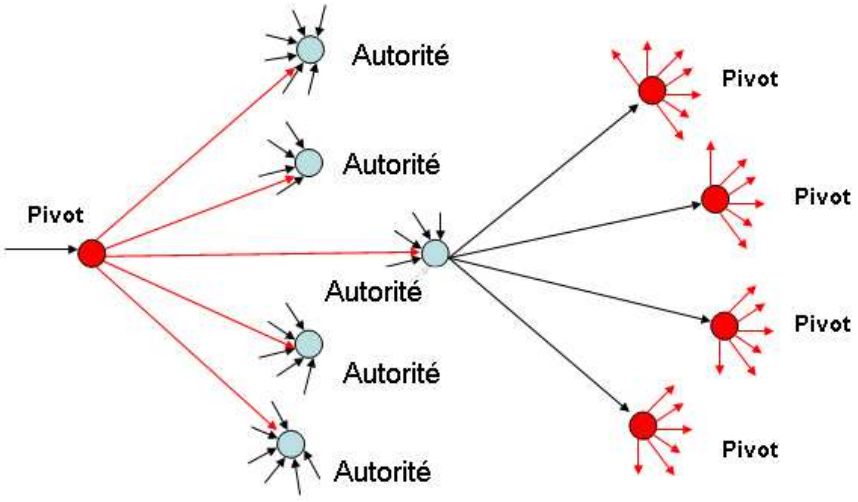


FIGURE 2.5: Exemple des pages pivots (Hubs) et autorités (Authorities)

rayonnantes (appelées ***"pages pivots"***) (voir la figure 2.5). Les estimations du degré de popularité et du degré de rayonnement des pages sont évaluées selon une logique circulaire : une page est d'autant plus populaire qu'elle est citée par des pages rayonnantes et réciproquement, une page est d'autant plus rayonnante qu'elle cite des pages populaires. Appliqué sur un ensemble de pages obtenues à la suite d'une requête sur un moteur de recherche, le système Clever met en œuvre une heuristique itérative faisant converger ces estimations vers des valeurs finales permettant d'identifier et de classer les pages, soit par ordre de popularité, soit par ordre de rayonnement. C'est de l'algorithme de HITS de Kleinberg [Kle99] qu'il s'agit. L'idée de Kleinberg trouve son origine dans des recherches beaucoup plus anciennes en bibliométrie : la méthode de Pinski-Narin [PN76] pour évaluer le poids d'une publication scientifique en fonction du nombres d'autres publication qui la cite. Si l'on observe la structure des liens, une page pivot cite des pages autorités, tandis qu'une page autorité est cité par des pages pivots.

Kleinberg a présenté une procédure pour identifier les pages Web qui sont des bonnes pages pivots ou des bonnes pages autorités, en réponse à une requête utilisateur donnée. Il s'avère utile de distinguer les pages pivots des pages autorités. Les premières correspondent aux pages possédant un nombre important de liens sortants (comme, par exemple, la page d'accueil de Yahoo !). Généralement, ces pages pivots sont des index. Tandis que les secondes correspondent aux pages beaucoup citées par d'autres pages. L'hypothèse stipule : *"Une page qui cite beaucoup de bonnes pages autorités est une bonne page pivot, et une page citée par beaucoup de bonnes pages pivots est une bonne page autorité"*. Pour identifier ces bonnes pages pivots et autorités, l'algorithme de HITS utilise la structure du Web vu comme graphe orienté. Étant donné une requête Q, le procédé de Kleinberg construit d'abord un sous-graphe G construit de pages contenant les termes de la requête, ainsi que les pages citées et citant ces pages. Puis il calcule un poids pivot et un poids autorité de chaque nœd du G (soit n nœds au total). L'algorithme HITS distingue deux types de liens (inter site et intra site). Les liens inter site établissent des relations entre des pages appartenant à des sites différents et pouvant être vus comme des liens de proximité sémantique entre les pages Web. Tandis que les liens intra site établissent des relations entre des pages d'un même site dont le premier but est de faciliter la navigation à l'intérieur d'un site (liens de navigation définissant la structure d'un site). Tous les liens intra site ont été supprimés du graphe en gardant que les liens inter site. Nous décrivons maintenant brièvement comment ces poids sont calculés :

Supposant W la matrice d'adjacence du sous-graphe orienté G. Notons respectivement par X et Y les deux vecteurs colonnes pivot et autorité de dimension $(n * 1)$ contenant les poids pivots et autorités correspondant à chaque nœud du sous-graphe G. Kleinberg utilise un processus itératif afin de calculer ces poids. Le poids

autorité du nœud *i*, X_i, est égale à la somme des poids pivots de tous les nœuds citant le nœud *i* et, pareillement, le poids pivot du nœud *i*, Y_i, est égale à la somme des poids autorités de tous les nœuds que cite le nœud *i*. Les poids pivots et autorités sont calculés de la façon suivante :

$$X_i^{(k+1)} = \sum_{j:j\to i} Y_j^{(k)} \quad et \quad Y_i^{(k+1)} = \sum_{j:i\to j} X_j^{(k)} \tag{2.3}$$

L'un des avantages de l'algorithme de HITS est le double classement. En pratique, le résultat de HITS est composé de deux listes ordonnées : une liste de bonnes pages autorités et une autre de bonnes pages pivots qui seront renvoyées à l'utilisateur. L'utilisateur a l'embarras du choix entre les deux listes et il peut être intéressé par une liste au détriment de l'autre selon la recherche demandée. Cependant, il existe quelques inconvénients de l'algorithme HITS. Les trois soucis majors que souffre l'algorithme HITS sont :

- La dépendance de HITS par rapport à la requête utilisateur. En effet, à l'interrogation du système, un graphe de voisinage doit être établi pour chaque requête exécutée. Or, le nombre d'itérations nécessaires à la determination des différents scores autorités et pivots est proportionnel à l'ensemble de réponses à une requête donnée. De ce fait, le temps de réponse est élevé pour des ensembles de réponses volumineux.
- L'algorithme de HITS est vulnérable aux techniques de spamming de lien. En ajoutant des liens artificiels vers les documents qui font autorité, un utilisateur peut influencer le poids autorité et le poids pivot de son document. Un léger changement de ces poids pourrait déplacer le document Web vers le top de la liste retournée à l'utilisateur. Avec des bannières publicitaires et des opportunités de financement, les propriétaires d'un document Web auront l'intention d'améliorer leur réputation dans le top du classement de la liste à retourner à l'utilisateur. Ceci est spécialement important dans la RI puisque l'utilisateur consulte seulement les 10 à 20 premières pages dans la liste des résultats. Dans l'esprit du propriétaire du document Web, ajouter des liens qui partent de son document est plus facile que d'ajouter des liens qui pointent vers son document. Ainsi, influencer les poids pivots des documents n'est pas difficile et comme les poids autorités et les poids pivots sont mutuellement dépendants, le poids autorité d'un document augmente quand son poids pivot augmente. En outre, puisque le graphe de voisinage est petit par rapport au Web entier, les changements locaux à la structure des liens s'avèrent affecter les calculs.
- Un autre inconvénient de HITS est le problème du sens des thèmes. En établissant le graphe G de voisinage pour une requête donnée, il est possible qu'un document de très bonne autorité non pertinent à la requête cite des documents contenant les termes de la requête. Ce document de très bonne autorité peut propager autant de poids que lui et ses documents voisins dominent la liste des documents pertinents renvoyés à l'utilisateur. Henzinger et Bharat [BH98] suggèrent une solution au problème du sens des thèmes, en pondérant les poids autorités et pivots des nœuds dans le graphe G par une mesure de pertinence par rapport à la requête. En fait, pour mesurer la pertinence d'un nœud dans le graphe G (i.e., un document) par rapport à la requête, ils utilisent la mesure de similarité cosinus.

2.2.2.2 Salsa

Un algorithme alternatif, The Stochastic Approach for Link-Structure Analysis (SALSA), est proposé par Lempel et Moran [LM00] qui combine les deux idées de HITS (Kleinberg [Kle99]) et PageRank (Brin et al. [BP98]) . Comme dans le cas de HITS, le graphe du Web est vu comme un graphe biparti (voir la figure 2.6), où les documents pivots pointent vers les documents autorités. L'algorithme de SALSA applique le principe de surfer aléatoire de PageRank sur le graphe biparti en alternant entre le coté pivot et le coté autorité. Le parcours aléatoire commence à partir d'un certain document autorité choisi aléatoirement. Le principe est le suivant : Quand on est sur un document du côté autorité du graphe biparti, l'algorithme choisit un des liens entrants aléatoirement qui pointe vers un document pivot du côté pivot et quand on est sur un document du côté pivot du graphe biparti, l'algorithme choisit un des liens sortants aléatoirement qui pointe vers un document autorité du

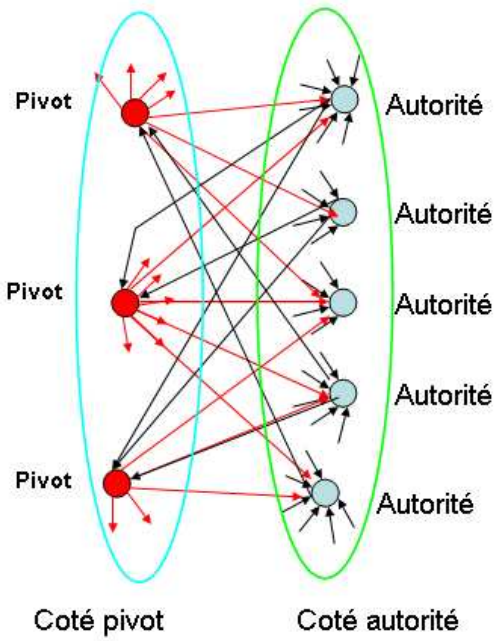


FIGURE 2.6: Graphe biparti de SALSA

coté autorité du graphe. Les poids autorités sont définis pour être la distribution stationnaire du surfer aléatoire. La formule de SALSA est définit comme suit :

$$X_i^{(k+1)} = \sum_{j:j\to i} \frac{1}{s_j} Y_j^{(k)} \quad et \quad Y_i^{(k+1)} = \sum_{j:i\to j} \frac{1}{e_j} X_j^{(k)} \tag{2.4}$$

Où e_i et s_i représente le nombre de liens entrants et sortants de la page *i* respectivement. Comme SALSA a été développé en combinant certaines des meilleures caractéristiques des deux algorithmes HITS et PageRank, il a beaucoup d'avantages. Différents de HITS, SALSA n'est pas victime du problème de sens des thèmes. En plus, rappelons que le problème de HITS était sa vulnérabilité au spamming des liens dû à la dépendance des poids pivots et d'autorités. SALSA est moins vulnérable au spamming des liens parce que le couplage est beaucoup moins important entre les poids pivots et les poids autorités. Cependant, ni HITS ni SALSA ne sont d'ailleurs imperméables au spamming des liens. L'inconvénient principal de SALSA est sa dépendance à la requête utilisateur comme le cas de HITS. Car, lors de l'interrogation, un graphe de voisinage doit être créé et deux vecteurs associés aux poids pivots et autorités doivent être calculé pour chaque requête.

2.3 Propagation de pertinence

Notre travail porte sur la notion de la propagation de pertinence. Afin de comprendre notre démarche, une études de différents systèmes qui reposent sur la propagation de pertinence s'avère utile. Nous allons vous presenter dans ce qui suit quelques travaux de recherche reposant sur la propagation de pertinence dont lesquels notre système est inspiré. Nous portons une attention particulière aux approches génériques mettant en oeuvre la technique de propagation de pertinence à travers le graphe du Web, en insistant sur les paramètres de propagation utilisés par ces travaux.

La propagation de pertinence améliore la propagation de popularité en prenant en compte la pertinence des documents : ***"un document cité par un grand nombre de documents pertinents à la requête utilisateur***

est un bon document". Les méthodes de cette catégorie de propagation de pertinence reposent sur l'amélioration de la pertinence d'un document en fonction de la pertinence de ces documents voisins. La propagation se fait du document source du lien vers le document destinataire, ou dans l'autre sens, ou encore en combinant les deux possibilités. On distingue deux variantes de ces méthodes : celles qui propagent la pertinence d'un document par rapport à la requête utilisateur à travers le graphe des liens (Frisse [FS92], Savoy [SR00], Marchiori [Mar97], Frei et al. [FS95], Shakery et al.[SZ03]), et celles qui propagent la pertinence à travers la structure hiérarchique des sites Web [WSC$^+$03]. Cette dernière méthode consiste à propager de la pertinence uniquement de bas vers le haut dans la structure arborescence d'un site. La propagation de pertinence permet de réduire considérablement la taille du graphe sur lequel est appliqué l'algorithme de propagation. En effet, un sous-ensemble de documents répondant à la requête est créé pour chaque requête exécutée dans le système. Plusieurs techniques de propagation de pertinence ont été proposées pour améliorer les performances de la recherche d'information sur le Web. Dans cette section, nous étudions brièvement quelques méthodes qui portent sur la propagation d'une fraction du score de pertinence à travers le graphe du Web. Puis, nous exposons un cadre général de propagation de pertinence proposé par Shakery et al. [SZ03] sur lequel reposent beaucoup de travaux de propagation de pertinence existants.

2.3.1 Propagation d'une fraction du score de pertinence

Les premières applications sur la propagation d'une fraction du score de pertinence sont l'œuvre des travaux de Frisse [Fri87] sur la recherche médicale. En effet, Frisse [Fri87] a proposé de tenir compte des liens hypertextes dans un système médical afin d'améliorer à la fois la pertinence des documents médicaux et la performance de la système de recherche documentaire. Il a proposé une heuristique simple de propagation de pertinence dans un système hypertexte dont la pertinence d'un document dans le système hypertexte est calculée en fonction du score de pertinence du document lui-même par rapport à la requête et de la pertinence de ces documents voisins.

D'autres travaux de propagation de pertinence appliqués à des bases documentaires ont été proposés par Savoy et al. [SR00]. Ces derniers ont analysé les réferences bibliographiques présents dans les articles scientifiques [Gar83]. L'hypothèse sous-jacente des travaux de Savoy et al. [SR00] repose sur la notion de la validité des references bibliographiques. En effet, le but principal d'une référence bibliographique est de citer des travaux antérieurs (modèles, méthodologies, résultats, etc.) en relation étroite avec le sujet de la publication. Cette hypothèse semble être valide pour la majorité des liens de références bibliographiques mais elle n'est pas vérifiée pour tous les liens [Liu93]. Le principe de l'algorithme proposé par Savoy et al. [SR00] repose sur l'analyse des liens de references bibliographiques des *M* premiers documents retrouvés par un moteur de recherche standard basé sur le contenu seul des documents. L'algorithme est décrit comme suit : s'il existe un lien de référence entre un document D_i et un autre document D_j, une fraction du score de pertinence du document $D_i \alpha * SD(D_i, Q)$ (généralement α est compris entre 0.1 et 0.2) est ajoutée au score de pertinence du document D_j ($SD(D_j, Q)$). La procedure de propagation de pertinence est répétée *C* fois. Le score de pertinence final d'un document est une combinaison de son score reposant sur le contenu seul et de la propagation des fractions des scores de ces documents voisins. la valeur de pertinence d'un document D_i : $RSV(D_i)$ (Relevance Status Value) est définit comme suit :

$$RSV\left(D_i\right) = SD\left(D_i, Q\right) + \alpha \cdot \sum_{j=1}^{M} SD\left(D_j, Q\right) \tag{2.5}$$

L'algorithme décrit ci-dessus possède quelques variantes :

- accorder un poids identique à tous les liens ;
- pondérer chaque lien individuellement selon son type (entrant ou sortant).

Une autre démarche similaire à celle de Savoy, mais appliquée sur le Web, à été proposée par Marchiori [Mar97] dans son moteur de recherche *"Hyper Search Engines"*. En effet, en plus du contenu textuel des pages, les liens entrants ou sortants peuvent fournir des indications précieuses afin de revoir le classement des pages retournées à l'utilisateur. Par exemple, la visibilité d'une page se mesure par le nombre de liens pointant vers cette page et correspond à une indication de sa valeur. Cependant, cette indication ne fournit pas directement une valeur sur le contenu informationnel de la page et il ne serait pas judicieux d'établir un lien entre la popularité d'une page et sa qualité. Les liens ne doivent fournir qu'une indication secondaire (surtout les liens entrants) et le contenu textuel des documents reste prépondérant dans le calcul du score de pertinence. De plus, l'influence d'une page sur une autre page diminue exponentiellement en fonction du nombre de liens qui les séparent selon la formule suivante :

$$SD(D_i, Q) = \sum_{D_j \in V_k(D_i)} F^k * SD(D_j, Q) \tag{2.6}$$

Avec $V_k(D_i)$ représente le voisinage de degree *K* du document D_i. $V_k(D_i)$ est composé de l'ensemble de documents *D* dont le nombre de liens qui séparent les deux documents *D* et D_i vaut *K*. Les caractéristiques de l'algorithme de Marchiori [Mar97] sont :

- Application de la propagation des scores de pertinence pour les 100 premiers documents retrouvés.
- Ne pas tenir compte des liens entrants.
- Pondération des liens sortants d'un facteur fixe de 0.75.

Frei et al. [FS95] ont proposé une méthode discriminatoire d'utilisation des liens. Au lieu de traiter tous les liens entre documents de la même façon, Frei et al. [FS95] ont annoté chaque lien avec une description de contenu spécifique. Ce contenu spécifique est composé de termes communs entre le document source et le document destinataire du lien. Ils ont suivi uniquement les liens dont la similarité entre la description de la requête et la description du lien est supérieure à un seuil fixe. La valeur de pertinence d'un document est mis à jour chaque fois que le lien est traversé. Elle est alors fonction de la valeur de pertinence du document par rapport à la requête et des valeurs de pertinence des documents liés à ce document. Le calcul de RSV, comme décrit dans l'équation 2.5, est non seulement semblable essentiellement à la stratégie d'enrichissement d'un document reposant sur les liens et proposée par Marchiori [Mar97] mais ressemble également sous une forme aux algorithmes d'analyse de liens qui propagent de l'information à travers les liens hypertextes. Les différents travaux de propagation d'une fraction du score de pertinence des documents ont les mêmes caractéristiques suivantes :

- La propagation de pertinence est appliqué à un ensemble réduit composé de documents pertinents à la requête. Dans la plupart des cas, ils prennent en considération que les K premiers documents les plus pertinents à la requête utilisateur.
- Le paramètre de propagation de pertinence est statique et compris entre 0 et 1.
- Les liens entrants et sortants sont pondérés selon leur type (entrant ou sortant) ou en fonction de la similartié entre le document source et le document destinataire du lien.

Dans la section suivante, nous présentons le modèle générique de propagation de pertinence proposé par Shakery et al. [SZ03] et repose sur le comportement de l'utilisateur pendant sa recherche.

2.3.2 Modèle général de propagation de pertinence

Shakery et al. [SZ03] ont proposé un cadre général de la propagation de pertinence des algorithmes existants. Ils distinguent deux facteurs importants pour le calcul de la pertinence d'un document : le score de pertinence du document lui-même et les scores de pertinence de ces voisins (documents qui pointent ou pointés par ce document). En effet, Shakery et al. [SZ03] propagent le score de pertinence d'un document à travers les liens hypertextes qui relient ces documents. Ils ont défini un score de pertinence pour chaque document (appelé score de pertinence sup) comme une fonction de trois variables : le score de pertinence du document

par rapport à la requête utilisateur, la somme des scores de pertinence des documents qui pointent le document courant et la somme des scores de pertinence des documents pointés par le document courant. Formellement, le modèle générique de propagation de pertinence proposé par Shakery et al. [SZ03] est décrit comme suit :

$$h^{k+1}(d,q) = f(S(d,q), \sum_{d_i \to d} h^k(d_i,q) \cdot w_I(d_i,d), \sum_{d \to d_j} h^k(d_j,q) \cdot w_O(d,d_j)) \tag{2.7}$$

Où $h^k(d,q)$ est le score de pertinence du document *d* par rapport à la requête *q* à la k^{me} étape du calcul, $S(d)$ est le score de pertinence du document *d* par rapport à la requête utilisateur *q* et w_I et w_O sont deux poids de pondération associés aux liens entrants et sortants du document *d*. Dans la pratique, ces poids sont calculés à partir du graphe des liens indépendamment des termes de la requête, généralement, on les fixe avant d'effectuer la propagation. Le choix de la fonction *f* qui combine les trois variables est aléatoire. Shakery et al. [SZ03] ont choisit la combinaison linéaire des trois variables ci-dessous :

$$h^{k+1}(d,q) = \alpha \cdot S(d,q) + \beta \sum_{d_i \to d} h^k(d_i,q) \cdot w_I(d_i,d) + \lambda \sum_{d \to d_j} h^k(d_j,q) \cdot w_O(d,d_j) \tag{2.8}$$

avec $\alpha + \beta + \lambda = 1$
Les scores de pertinence sont calculés calculés itérativement jusqu'à ce que l'algorithme converge à une limite qui n'est que le score final de pertinence des documents à classer. Dans la pratique, Shakery et al. [SZ03] distinguent trois cas simplifiés des modèles spécifiques de propagation de pertinence. Chaque cas correspond à un certain comportement de l'utilisateur pendant sa recherche.

2.3.2.1 Pondération des liens entrants

Ce modèle de comportement d'utilisateur est tout à fait semblable au modèle de PageRank (Brin et al. [BP98]), sauf qu'il n'est pas indépendant de la requête. Le principe de l'algorithme est le suivant : Soit un surfer aléatoire qui part d'un document au hasard de l'ensemble des résultats de la requête (mesure de contenu). A partir de ce document, le surfer aléatoire decide de suivre un des liens hypertextes qui se trouve dans le document courant et donc de visiter un document voisin. S'il juge que ce dernier est pertinent à la requête, il continue de naviguer dans l'espace du travail en suivant le même procédé lorsque il est dans le premier document. Lorsqu'il juge que le contenu du document courant est non pertinent à la requête utilisateur, il saute vers un autre document de l'ensemble du travail tiré au hasard. Puis il recommence sa navigation à partir d'un document aléatoire. La probabilité que le surfer aléatoire visite un document représente le score de pertinence de ce document. Ce modèle a certaines caractéristiques qui le distinguent de PageRank. D'abord, il est appliqué à un sous-ensemble de documents, qui constitue l'ensemble du travail, plutôt que d'être appliqué à la collection entière. Une des caractéristiques de cet algorithme est que l'ensemble du travail est composé des documents dont le score de pertinence des documents par rapport à la requête utilisateur est supérieure à un seuil. En second lieu, la probabilité que le surfer aléatoire visite un lien est proportionnelle à la pertinence du document destinataire du lien. Il est fort probable que l'utilisateur visite un lien qui mène vers un document pertinent à la requête utilisateur que vers un document non pertinent à la requête utilisateur. Ce comportement d'utilisateur peut être formulé comme suit où à chaque itération de l'algorithme de propagation, un nouveau score de pertinence sup pour chaque document sera calculé de la manière suivante :

$$h^{k+1}(d) = \alpha \cdot S(d) + (1-\alpha) \sum_{d_i \to d} h^k(d_i) \cdot w_I(d_i,d) \qquad (0 < \alpha < 1) \tag{2.9}$$

2.3.2.2 Pondération des liens sortants

Dans ce modèle, l'utilisateur a le choix de lire le contenu du document avec une probabilité α ou de visiter l'un de ces documents voisins en suivant un lien hypertexte avec une probabilité de $(1-\alpha)$. Avec ce modèle,

le calcul des scores de pertinence des documents est décrit par la fonction suivante :

$$h^{k+1}(d) = \alpha \cdot S(d) + (1-\alpha) \sum_{d \to d_j} h^k(d_j) \cdot w_O(d, d_j) \qquad (0 < \alpha < 1) \tag{2.10}$$

A chaque itération, le score de pertinence sup de chaque document est calculé en combinant le score de pertinence basé sur le contenu seul du document par rapport à la requête avec les scores de pertinence sup des documents pointés par ce document. Ces derniers n'ont pas le même impact sur le score de pertinence du document. En effet, le poids d'un liens sortants $w_o(d \to d_J)$ est calculé en fonction de la pertinence du document destinataire du lien.

2.3.2.3 Pondération uniforme des liens sortants

Dans ce cas particulier, on suppose que chaque fois qu'un document est visité, l'utilisateur lit son contenu intégralement avant de passer à un de ces documents voisins. La probabilité de suivre un lien vers un document voisin est de $(1-\alpha)$. La formule de calcul des scores de pertinence sup pour ce modèle est définie comme suit :

$$h^{k+1}(d) = S(d) + (1-\alpha) \sum_{d \to d_j} h^k(d_j) \qquad (0 < \alpha < 1) \tag{2.11}$$

Shakery et al. [SZ03] ont évalué les trois algorithmes issus du modèle générique sur la collection Gov avec deux ensembles de requêtes : celles de 2002 (Req_1) et celles de 2003(Req_2). Les résultats obtenus avec les requêtes Req_1 sont au-dessus de l'algorithme de base reposant sur le contenu seul des documents pour des paramètres α différents selon l'algorithme utilisé. En effet, la méthode de pondération uniforme des liens sortants donne de meilleurs résultats que l'algorithme de base quel que soit α, tandis que la méthode de pondération des liens sortants améliore les résultats pour un $\alpha > 0,2$ et celle de la pondération des liens entrants pour un $\alpha > 0,7$. Cependant, les résultats obtenus avec l'ensemble de requête Req_2 sont mauvais quelle que soit la valeur de propagation α et en tout mode de pondération des liens. On conclue que les méthodes du modèle générique de propagation étudiées auparavant ne sont performantes que dans des cas spécifiques et dépendent essentiellement de la collection et des paramètres utilisés. Les résultats obtenus lors des expérimentations montrent que l'amélioration de performance des système de recherche d'information reposant sur la propagation de pertinence est sensible au choix de la collection de document et des paramètres de propagation utilisés. Dans la pratique, il est difficile de d'appliquer ces modèle vu leur indépendance par rapport au facteur de propagation.

2.3.2.4 Propagation de pertinence à travers la structure physique du site

Wen et al. [WSC$^+$03] ont proposé un algorithme de propagation à travers la structure hiérarchique d'un site. D'abord, ils ont construit un arbre pour chaque site Web à partir des URLs des documents appartenant au site. Ensuite, ils ont propagé les fréquences des termes de la requête par rapport à la relation fils-père dans l'arbre du site. La fréquence d'occurrence d'un terme *t* dans ce modèle est décrite comme suit :

$$f_t^{`}(p) = (1+\alpha) f_t(p) + \frac{(1-\alpha)}{|Fils(p)|} \sum_{q \in Filles(p)} f_t(q) \tag{2.12}$$

Où $f_t^{`}(p)$ et $f_t(p)$ sont les fréquences d'occurrence du terme *t* dans le document *p* avant et après la propagation, q est la page fille de p. Après la propagation des fréquences de termes, n'importe quel algorithme de calcul de pertinence des documents peut être utilisé pour raffiner le classement des documents.

2.3.3 Propagation de pertinence probabiliste

Un autre modèle de propagation été proposé par Shakery et al. [SZ06] repose sur la propagation des probabilité de pertinence de documents à travers les liens hypertextes au lieu de la propagation des scores de pertinence. Il s'agit de calculer la probabilité qu'un document soit pertinent à la requête en fonction de son score de pertinence et des scores de pertinence des documents répondant à la requête. Puis de propager ces probabilités à travers les liens hypertextes. Shakery et al. [SZ06] distinguent plusieurs ensembles de voisinage : voisinage des liens entrants, voisinage des liens sortants, voisinage composé du document lui même, voisinage composé de tous les documents de l'ensemble de travail, etc. Le principe du modèle probabiliste de propagation de pertinence est le même que le modèle du surfer aléatoire. La probabilité qu'un surfer visite un document D est définit comme suit :

$$\begin{aligned} P(d,q) &= \sum_{i=1}^{k} \alpha_i \sum_{d' \in C} p(d',q) p_i\left(d' \rightarrow d\right) \\ \sum_{i=1}^{k} \alpha_i &= 1, \qquad \sum_{d \in C} p_i\left(d' \rightarrow d\right) = 1 \end{aligned} \tag{2.13}$$

avec C l'ensemble de documents répondant à la requête utilisateur. α_i la probabilité de choisir un type particulier de voisinage du document lorsque le surfer aléatoire quitte le document courant et p_i est la probabilité de visiter une page particulière dans l'ensemble de voisinage choisit.

2.3.4 Discussion sur les modèles de propagation de pertinence

Les expérimentations effectuées avec une variante du modèle BM25 des différents systèmes de propagation de pertinence montrent une amélioration des résultats obtenus sur le Web track de TREC (Savoy et al. [SR00],Craswell et al. [CH04]). Cependant, cette amélioration dépend du paramètre de propagation de pertinence utilisé et de l'ensemble des requêtes exécutées. En effet, la plupart des modèles de propagation de pertinence étudiés dans cette section appliqués à des corpus de documents différents montrent que les performances de ces systèmes dépendent du paramètre de propagation de pertinence utilisé par le système qui est en général fixé par l'algorithme ou calculé à partir de la similarité entre les documents reliés. Or, ces modèles de propagation de pertinence ne tiennent pas compte du nombre de termes de la requête utilisateur dans le calcul de la pertinence des documents par rapport à la requête utilisateur. En effet, la taille du sous-graphe dépendant de la requête est proportionnel au nombre de termes de la requête utilisée. Plus la requête contient de termes, plus le nombre de réponses à la requête utilisateur est grand, d'où un nombre important de liens dans le sous-graphe. De ce fait, il ne faut pas donnée plus d'importance au voisinage des documents. Cette importance devrait dépendre du nombre de termes de la requête utilisateur. C'est ce que nous préposons dans notre démarche. Au lieu de fixer le paramètre de propagation, nous allons le calculer dynamiquement pendant l'interrogation du système en fonction du nombre de termes de la requête utilisateur. La nouveauté de notre approche de propagation de pertinence reside dans le choix du paramètre de propagation que nous allons vous le presenter dans la chapitre 3. Bien évidement, les techniques de propagation de pertinence ont des failles. Nous citons le problème du spamming de liens qui reste le problème majeur de tous les algorithmes d'analyse de liens en recherche d'information. En effet, il est facile de fausser les calculs de scores de pertinence des pages Web en ajoutant des liens entre les pages qui contiennent les termes de la requête. Les modèle de propagation de pertinence ne distinguent pas entre les liens vides et les liens informationnels puisque elles les traitent équitablement en fixant le paramètre de propagation. Le deuxième problème est le sens des thèmes. Les différentes techniques de propagation de pertinence étudiées dans ce chapitre considèrent la page Web comme l'unité d'information la plus petite à retrouner à l'utilisateur. Or, la plupart des pages Web traitent plusieurs sujets de thématiques différentes et les liens hypertextes ne pointent pas la page cible entière mais la partie de la page de la même thématique que la page source du lien.

2.4 Analyse des liens au niveau blocs thématiques

Dans le chapitre précédent, nous avons étudié quelques algorithmes d'analyse de lien au niveau page Web. Or, il s'avère que les deux hypothèses des algorithmes d'analyse des liens ne sont vérifiées dans la plupart des cas. La première hypothèse résulte d'une analogie entre la bibliométrie et le Web. Une des limites de cette analogie entre les deux domaines est de considérer tous les liens hypertextes comme des liens de citation ou de référence. En effet, dans le réseau des publications scientifiques [Gar83], un lien de référence est considéré comme une citation et indique une relation importante entre la page qui contient la référence bibliographique et la page référencée. Or, les documents Web contiennent de plus en plus de liens automatiques, et que certains d'entre eux sont liés à des objectifs publicitaires, commerciaux ou de navigation, et n'indiquent pas une pertinence réelle accordée par l'auteur du document aux documents cités. De plus, la course à la visibilité des documents ou sites Web et la concurrence entre les différents acteurs du Web ont poussé les auteurs de sites commerciaux et publicitaires à faire plus d'efforts pour rendre leur site visible dans les premières pages des résultats retournés par un moteur de recherche. De ce fait, tenir compte de ces liens vides d'information (les liens de publicité et les liens de navigation qui n'ont rien à voir avec le rapprochement sémantique entre les pages Web) dans le calcul de la pertinence des documents par rapport à la requête utilisateur peut introduire du bruit et par conséquent de dégrader les performances de la recherche. La deuxième hypothèse des algorithmes d'analyse de liens repose sur la granularité de l'information à retourner à l'utilisateur du système qui est la page Web. En effet, la plupart des applications Web considèrent la page Web en tant que la plus petite unité d'information indivisible. Le calcul de la pertinence se fait au niveau page. Cependant, une page Web contient souvent divers contenus de thématiques différentes. Ainsi, une page Web ne devrait pas être la plus petite unité d'information et par conséquent, la détection des structures de contenu thématique d'une page devient un facteur potentiel pour l'amélioration des performances de la recherche d'information sur le Web.

C'est pour ces deux raisons que beaucoup de travaux de recherche sont portés sur le découpage relatif ou absolu des pages Web en blocs thématiques et ont calculé de la pertinence au niveau bloc Cai et al. [CYWM04]. Le découpage relatif des pages Web consiste à identifier les différents thèmes de la page en utilisant par exemple la classification des termes contenus dans la page Nie et al. [NDQ06] ou les relations existantes entre les pages Web et d'autres pages dont on connaît la thématique tel que les annuaires[Hav02]. Tandis que le découpage absolu des pages Web repose sur la segmentation physique des pages Web. Le but est d'arriver à découper une page en blocs d'informations différents. Par définition, la segmentation est une tâche qui vise à déterminer une segmentation thématique a priori, indépendamment de toute requête. Il s'agit plus précisément de découper le contenu textuel d'une page en une succession de segments thématiquement homogènes, de caractériser ces segments en termes de contenu, et éventuellement d'établir certaines formes d'organisation reliant ces segments. Différentes méthodes d'analyse thématique automatique ont été proposées dans le passé. Nous distinguons deux grandes familles de segmentation :

- Les méthodes qualifiées de quantitatives ou numériques reposent sur la notion de cohésion lexicale (Halliday et al. [HH76]), en exploitant la répétition des mots comme indicateur d'homogénéité thématique. Ces méthodes procèdent à une segmentation linéaire du texte, c'est-à-dire en segments adjacents.
- Les méthodes qualifiées de linguistiques exploitent des critères linguistiques (formes linguistiques des mots (verbe, sujet, ponctuation, etc), syntaxe de la langue, etc) afin de découper les pages en blocs thématiques.

Dans ce qui suit, nous nous intéressons qu'aux méthodes quantitatives pour deux raisons :

- Ces méthodes sont simple à mettre en oeuvre. En effet, elles reposent sur des calcul numériques de similarité entre les segments resultants. De plus, elles n'ont pas besoin d'aucune ressource externe comme les règles de la langue utilisée et ces formes linguistiques dans les calculs.
- Ces méthodes sont adaptées au Web vu le nombre important de pages que contient le Web. En effet, il est difficile de découper toutes les pages Web en tenant compte des critères linguistiques.

Et comme notre travail porte aussi sur la propagation de pertinence au niveau d'une unité d'information inférieure à celle de la page Web dite bloc thématique, une étude des algorithmes d'analyse de liens qui porte sur le calcul de la pertinence au niveau bloc thématique (physique ou relatif) s'avère nécessaire pour comprendre notre démarche. Avant de présenter ces travaux, un état de l'art sur les différentes méthodes de segmentation linéaire du texte brut et les méthodes de segmentation structurelle des pages Web les plus connues est introduit dans la section suivante. La premiere catégorie concernent les travaux de segmentation de textes en se basant sur la répétition des mots dans le texte et leurs occurrences. C'est le cas des travaux Morris et al. [MH91], Hearst [Hea97], Richmond et al. [RSA97] et Yaari [Yaa97] qui procèdent à un découpage linéaire du texte brut en blocs adjacent. La deuxième catégorie représentent les méthodes de segmentation des pages HTML appliquées sur le Web en utilisant la structure HTML dans le choix des délimiteurs de segments. Notre travail de segmentation de pages en blocs thématiques est inspiré de ces deux catégories. En effet, l'algorithme de segmentation que nous allons vous présenter en détail dans le chapitre 3 utilise à la fois les critères visuels et de présentation de la pages (structure HTML de la page i.e. <HR>, <P>, <H1>..<H6>,etc.) et une fonction d'évaluation des différentes solutions de segmentation reposant sur les mesures de similarité et de disimilarité entre les segments des méthodes de segmentation linéaire du texte brut. Dans ce qui suit, nous présentons d'abord les méthodes de segmentations, puis nous introduisons quelques algorithmes d'analyse de liens qui portent sur les thématiques relatives des pages et d'autres algorithmes d'analyse de liens qui calculent une certaine pertinence au niveau blocs thématiques et physiques.

2.4.1 Segmentation linéaire du texte brut par cohésion lexicale

Les approches de segmentation quantitatives ou numériques reposent sur la notion de cohésion lexicale (Halliday et al. [HH76]), en exploitant la répétition des mots comme indicateur d'homogénéité thématique. Il s'agit notamment des travaux se plaçant dans la lignée de Hearst [Hea94] [Hea97], Reynar [Rey94] ou encore Salton et al. [SSBM96] qui procèdent à une segmentation linéaire du texte c'est-à-dire en segments adjacents. Le principe commun à de nombreuses méthodes de cette famille est le suivant : un vecteur de termes (ex. mots bruts ou lemmatisés) est associé à chaque segment minimal (par exemple le paragraphe). Chaque composante du vecteur est une valeur numérique représentative de la fréquence du terme dans ce segment, généralement obtenue par TFIDF [SWY75]. Une fois le decoupage effectué, une mesure de distance vectorielle permet d'évaluer la cohésion thématique de chaque couple de segments. Ces derniers pourront alors être regroupés, par seuillage sur cette distance, en unités homogènes. Du point de vue de la caractérisation des segments, les termes fortement pondérés de chaque vecteur peuvent être utilisés pour caractériser le theme du segment. Dans ce qui suit nous présentons quelques algorithmes les plus connus de segmentation de texte par cohésion lexicale.

Morris et al. [MH91] décrivent un algorithme de segmentation du discours reposant sur les relations de la cohésion lexicale Halliday et al. [HH76]. Leur algorithme découpe le texte en segments selon une structure hiérarchique. La premiere étape de l'algorithme de Morris et al. [MH91] consiste à relier les occurrences des termes dans un document afin de former des chaînes lexicales. Deux termes forment une chaîne lexicale s'ils sont liés par une relation de cohesion lexicale. Chaque nouveau terme ajouté à la chaîne lexicale doit participer dans une relation de cohesion lexicale avec au moins un terme de la chaîne. Morris et al. [MH91] utilisent un thesaurus (Roget [Rog77]), composé de plusieurs catégories de mots et reliées par des liens, afin de determiner si une paire de termes satisfait l'une de ces relations de cohesion lexicale suivantes :

- Les deux termes appartiennent à la même catégorie.
- L'un des termes appartient à une catégorie qui contient un pointeur vers une catégorie contenant le deuxième terme.
- L'un des termes est un discriminer de la catégorie qui contient l'autre terme.
- Les deux termes appartiennent à deux catégories différentes qui pointent vers une catégorie en commun.

Une fois les chaînes lexicales sont identifiées dans un document, Morris et al. [MH91] comparent entre les différents elements des chaînes pour determiner s'elles vérifient continuité thématique.

Un autre algorithme de segmentation de texte par cohésion lexicale, plus connue sous le nom de TextTiling, est proposé par Hearst [Hea97] en 1997. L'algorithme de Hearst [Hea97] étudie la distribution des termes selon plusieurs critères. Un score de cohésion est attribué à chacun des blocs de texte en fonction du bloc qui le suit. Ce score dépend lui-même d'un second score attribué à chaque paire de phrases suivant la paire de phrases qui la suit. Ce second score est calculé en tenant compte des mots communs, du nombre de mots nouveaux, et du nombre de chaînes lexicales dans les phrases considérées. Le score d'un segment de texte est alors le produit scalaire normalisé des scores de chaque paire de phrases qu'il contient. Si l'écart entre le score d'un segment et les scores du segment qui le précède et du segment qui le suit est grand, une frontière est posée à l'intérieur de ce segment. La rupture entre deux unités thématiques est située dans une zone du texte entourée de zones présentant des valeurs de cohésion très différentes de la sienne. Ces ruptures sont visibles sur le graphe des valeurs de cohésion par des creux ou des bosses.

Richmond et al. [RSA97] définissent une technique de localisation de délimiteurs de segments thématiques dans un document. Ils calculent le poids de l'importance du terme dans un document en fonction de sa fréquence dans le document et de la distance entre les répétitions de ce terme. Ils déterminent la similarité entre deux blocs voisins par le rapport entre la somme des poids de termes en commun et la somme des poids de termes qui figurent uniquement dans l'un des deux blocs.

L'algorithme Segmenter, proposé par Kan et al. [KKM98], effectue une segmentation linéaire basée sur les chaînes lexicales présentes dans le texte. Une chaîne est rompue si le nombre de phrases séparant deux occurrences est trop important. Le nombre de phrases déterminant la coupure des chaînes dans Segmenter dépend de la catégorie syntaxique du terme considéré. Une fois tous les liens établis, un poids leur est assigné en fonction de la catégorie syntaxique des termes en jeu et de la longueur du lien. Un score est ensuite affecté à chaque bloc en fonction des poids et des origines des liens qui le traversent ou qui y sont créés. Les délimiteurs de la segmentation sont alors posées au début des blocs ayant les scores maximaux.

Yaari [Yaa97] propose que le texte soit découpé en utilisant le clustering hiérarchique HAC modifié. Au départ, il place chaque paragraphe dans une classe, puis récursivement, l'algorithme fusionne les paragraphes voisins les plus similaires dans une classe jusqu'à ce tous les blocs soient dans une même classe. La similarité entre les paragraphes est calculée en utilisant la mesure de cosinus avec l'IDF du document (inverse document frequency). Une fois le clustering hiérarchique des paragraphes obtenu, Yaari [Yaa97] applique des règles pour convertir le clustering hiérarchique resultant en une segmentation linéaire.

2.4.2 Segmentation structurelle de pages Web

De nombreux algorithmes ont été proposés pour segmenter les pages Web en segments cohérents de plus petite taille en tenant compte de la structure HTML de la page lorsque les découpages en paragraphes de l'auteur ne sont pas disponibles. Nous distinguons les trois catégories suivantes de méthodes de segmentation des pages Web :

2.4.2.1 Segmentation en blocs de taille fixe

Dans la recherche d'information traditionnelle, des passages de taille fixe ou fenêtre glissante ont été utilisées pour surmonter la difficulté de normalisation de la longueur des pages. Un passage de taille fixe est un bloc composé d'un nombre fixe de termes. Callan [Cal94] propose une approche de fenêtre glissante, dans laquelle la première fenêtre du document commence à la première occurrence d'un terme de la requête exécutée. Dans la segmentation de page en blocs de longueur fixe, la structure HTML du document n'est pas prise en compte.

En effet, l'algorithme de segmentation est appliqué au contenu textuel de la pages en supprimant toutes balises HTML de la page et leurs attributs . La longueur de la fenêtre glissante est le seul paramètre de l'algorithme de Callan [Cal94] fixé à 200 ou 250 termes dans les experimentations effectuées. La segmentation de page en blocs de longueur fixe est simple à mettre en oeuvre et elle peut s'avérer utile pour améliorer la précision de la recherche, particulièrement pour des collections contenant des documents courts et longs en même temps (Callan [Cal94], Kaszkiel [KZ01]). Le seul inconvenient de cette méthode est qu'aucune information sémantique n'est prise en compte dans le processus de segmentation.

2.4.2.2 Segmentation reposant sur l'arbre DOM (Document Object Model) des pages Web

Certains chercheurs ont essayé d'utiliser des techniques de base de données objets pour structurer les données du Web Hammer et al. [HGmC+97], Adelberg [Ade98], Ashish et al. [AK97]. Un des outils d'extraction de structures de documents, NoDoSE, a été proposé par Adelberg [Ade98]. NoDoSE représente une approche interactive de découverte de structures des pages Web. Dans NoDoSE, l'utilisateur identifie au départ quelques segments intéressants dans une page manuellement. L'algorithme cherche ensuite d'identifier d'autres segments intéressants de la même page en la décomposant d'une manière top-down et en utilisant la structure DOM des pages HTMLs. La structure découverte de cette page sera alors utilisée pour extraire des segments appropriés de pages Web de même structure. Bien que cette démarche soit intéressante, le besoin de l'intervention étendue de l'utilisateur peut s'avérer pénible, notamment en traitant un nombre important de pages fortement hétérogènes, comme le cas du Web. D'autres travaux antérieurs Embley et al. [EJN99], Chen et al. [CZS+01], Chakrabarti [Cha01], Chakrabarti et al [CPS02] ont utilisé l'arbre DOM d'une page HTML pour extraire des informations structurelles de la page afin de la découper en plusieurs blocs homogènes. Cependant, l'arbre DOM est initialement introduit pour la présentation des pages dans un navigateur plutôt que la description de sa structure sémantique. Un exemple frappant de la non pertinence de la structure DOM est la multi-fonctionnalité des balises HTML. Par exemple, la balise <TABLE> peut être utilisée comme un tableau de données ou un moyen de représentation de la structure sémantique d'une page Web.

2.4.2.3 Segmentation à critères visuels

Cai et al. [CYWM03] ont proposé un algorithme de segmentation reposant sur des critères visuels VIPS (VIsion-based Page Segmentation). Dans Cai et al. [CYWM03], les segments d'une page Web sont pré-définis. On distingue par exemple des blocs publicitaires, des blocs de navigation et des blocs de contenus dont l'emplacement de ces blocs dans la page est définit comme un standard. C'est le cas de plusieurs pages Web qui adaptent le même style de représentation des informations. Cai et al. [CYWM03] distinguent 4 blocs différents selon son emplacement dans la page Web : le bloc en haut de la page contient la plupart des blocs publicitaires, le bloc à droite de la page représente le menu de la page qui contient des liens de navigation à l'intérieur du site de la page, le bloc qui se trouve à gauche de la page contient des liens hypertextes vers des pages en relation avec la page courante (des liens informationnels) et enfin le reste du document qui représente le contenu du document. Donc il suffit de délimiter ces segments à l'aide des critères visuels pour séparer les quatre blocs pré-définis. L'algorithme VIPS vise à extraire la structure sémantique d'une page Web en se basant sur sa présentation visuelle. Une telle structure sémantique est une structure arborescente où chaque nœud dans l'arbre correspond à un bloc. Chaque nœud sera assigné une valeur qui représente le degré de cohérence du contenu du bloc. L'algorithme de VIPS extrait d'abord tous les blocs souhaitables à partir de l'arbre DOM d'une page HTML. Puis, il cherche des séparateurs entre les éventuels blocs. Les séparateurs utilisés dans la segmentation sont les lignes horizontales <HR> ou verticales contenues dans la page Web et la couleur du fond du texte. En se basant sur ces séparateurs, l'arbre sémantique de la page Web est construit. Le problème de cette approche réside dans le choix de séparateurs de segments. En effet, il existes plusieurs critères visuels candidats à la

segmentation. En plus, les blocs ne sont pas pré-définis. On peut trouver par exemple un bloc publicitaire en haut, à gauche et à droite d'une page Web. D'où un problème major de distinction entre les différents segments.

2.4.3 Utilisation de liens au niveau blocs

On distingue deux catégories de travaux qui tiennent compte des liens hypertextes dans le calcul d'un score de pertinence d'une page par par rapport à la requête utilisateur en fonction des thématiques de la page. La première catégorie est représentée par les travaux de Haveliwala [Hav02], Pal et al. [PND05] et Nie [NDQ06] qui ont proposé des approches qui consistent à calculer plusieurs scores de pertinence d'une page par rapport aux différentes thématiques abordées dans la page. Les blocs thématiques n'existent pas physiquement. Cependant, la deuxième catégorie est représentée par les deux algorithmes d'analyse de liens Block Level PageRank et Block Level Hypertext-Induced Topic Selection proposés par Cai et al. [CYWM04] qui reposent sur le découpage physique des pages Web en blocs. Ces deux derniers algorithmes ne calculent pas la pertinence au niveau bloc, mais ils font la distinction entre les blocs informationnels et les blocs vides tels que les blocs publicitaires et les blocs navigationnels. Nous détaillons ces différentes approches dans ce qui suit.

2.4.3.1 Utilisation de liens au niveau blocs thématiques relatifs

Dans cette section, nous présentons différentes adaptations de l'algorithme d'analyse de liens du PageRank afin de tenir compte de la thématiques des pages et les thématiques abordés dans une page sans faire appel au découpage physique des pages Web. Parmi ces travaux, on trouve le PageRank sensible à la thématique proposé par Haveliwala [Hav02] qui consiste à une réutilisation raffinée du PageRank traditionnel. L'idée de l'algorithme part d'un constat simple : ***Le PageRank permet de classer par ordre d'importance des pages répondant à un besoin utilisateur. Mais il ne permet pas de distinguer une page qui parle de l'animal jaguar ou de la voiture jaguar***. Donc, la solution à ce problème est de biaiser le calcul du PageRank en donnant plus d'importance aux pages bénéficiant de liens en provenance de sites dont on connaît la thématique du départ. Cet algorithme étend l'idée de PageRank originale en rajoutant un ajustement sensible à la thématique pendant la phase d'interrogation du système. En effet, au lieu de calculer un seul vecteur du PageRank, l'algorithme calcule une multitude de vecteurs partiels spécifiques pour chaque thème possible de l'ensemble des pages indexées. Cependant, la création de ces vecteurs du PageRank partiels nécessite des ressources considérables (sources thématiques qui peuvent être utilisées dans le but d'assigner une thématique pour chaque page de l'index).

Ainsi, dans la pratique, l'algorithme de PageRank sensible à la thématique utilise seulement les 16 vecteurs spécifiques du PageRank thématique représentant les différentes catégories de niveau supérieur du l'annuaire ODP (Open Directory Project). Le choix des catégories d'ODP comme étant des classes thématiques de pages Web est motivé par le fait que l'annuaire ODP est créé et révisé par un grand nombre d'utilisateurs volontaires et indépendants.

Le processus de fonctionnement d'un tel système est le suivant : pour chaque page Web, un ensemble de scores d'importances concernant les différents thèmes des catégories d'ODP est pré-calculé. Dans la phase d'interrogation, le score spécifique de chaque thème est combiné avec d'autres scores (par exemple le contenu textuel de la page) pour former le classement final de la page. Le résultat de l'algorithme est forcément composé de 17 vecteurs de PageRank différents : le vecteur de PageRank global, et les 16 vecteurs de PageRank partiels correspondant à chacune des catégories de niveau supérieur d'ODP. Le PageRank correspondant à la thématique de l'une des catégorie de niveau supérieur d'ODP est calculé de la manière suivante : Soit c_j une des 16 catégories de niveau supérieur d'ODP. Pour chaque thème c_j, il est nécessaire de calculer un vecteur de PageRank partiel. Soient p une page, $O(p)$ l'ensemble de pages citées par la page p et $T(c_j)$ l'ensemble

de pages de la catégorie c_j. $|O(p)|$ et $|T(c_j)|$ représentent le nombre de pages des ensembles $O(p)$ et $T(c_j)$ respectivement.Le score de PageRank correspondant au thème de la catégorie c_j de la page *p* est définit comme suit :

$$TSPR_{c_j}(p) = (1-d)\sum_{r:r\to p}\frac{TSPR_{c_j}(r)}{|O(r)|} + \begin{cases} \frac{d}{|T(c_j)|} & \text{si } p \in T(c_j) \\ 0 & \text{si } p \notin T(c_j) \end{cases} \tag{2.14}$$

Où $r \to p$ signifie qu'il existe un lien hypertexte partant de la page *r* vers la page *p*.

Afin d'utiliser les valeurs de PageRank partiel pendant l'interrogation du système, il suffit d'être en mesure de déterminer à quelle thématique se rattache une requête donnée. Pour découvrir la thématique associée à une requête utilisateur, deux scénarios possibles sont considérés : Dans le premier scénario un utilisateur met en valeur un mot clé présent dans une page *p* et lance sa requête. Dans ce cas de figure, le thème de la requête est défini par le contenu de la page *p* (ç.à.d la thématique de la page *p*). Par exemple si le mot ***'architecture'*** est mis en valeur dans une page concernant le domaine du bâtiment, les pages relatives à ***l'architecture des ordinateurs*** ne devraient pas apparaître parmi les résultats de la requête. Ainsi si un terme de la requête *q* est mis en valeur dans une page *p*, le contexte de la requête *q* serait constitué de termes présents dans la page *p*. Dans le deuxième scénario un utilisateur saisit des mots clés comme dans le cas d'un moteur de recherche standard. En effet, le contexte de la requête *q* est composé des termes de la requête elle-même. Afin de déterminer la thématique de la requête, il suffit de construire, pour chaque catégorie thématique c_j, un vecteur de termes D_j. Chaque élément du vecteur de termes D_j représente le nombre d'occurrences d'un terme dans l'ensemble de pages associées à la catégorie thématique c_j. La thématique associée à la requête utilisateur sera évaluée selon une méthode probabiliste. Pendant la phase d'interrogation du système, la proximité du contexte de la requête $q^{'}$ par rapport à l'un des thèmes c_j peut être calculée de la manière suivante :

$$P\left(c_j|q^{'}\right) = \frac{P\left(c_j\right)\cdot P\left(q^{'}|c_j\right)}{P\left(q^{'}\right)} \quad \propto \quad P\left(c_j\right)\cdot\prod_i P\left(q_i^{'}|c_j\right) \tag{2.15}$$

La valeur de proximité $P(q_i^{'}|c_j)$ du terme *i* de la requête *q* par rapport à la catégorie c_j est calculée en utilisant les termes du vecteur D_j associé à la catégorie c_j. Le score d'importance sensible à la thématique d'une page *p* qui contient les termes de la requête original *q* est calculé comme suit :

$$S_q(p) = \sum_j TSPR_{c_j}(p) * P(c_j|q^{'}) \tag{2.16}$$

Les résultats d'une requête particulière sont classés selon les valeurs de la fonction 2.16 qu'on vient de décrire.

Pal et al. [PND05] ont adapté le modèle de PageRank sensible à la thématique proposé par [Hav02] au moment de suivre un lien hypertexte en intégrant des informations sur la continuité thématique dans le graphe des liens tirée de l'historique des pages visitées par un surfeur sur le Web. Le principe de cet algorithme est le suivant :***"Au lieu de considérer les liens sortants uniformément, un surfeur visitant une page de thématique*** c_j ***préfère suivre que les liens qui citent des pages de la même thématique que la page visitée"***. Les pages de thématiques différentes que celle de la page visitée peuvent être visitées de temps en temps, mais avec une probabilité faible. Ce modèle ne prend pas en considération le principe du modèle du surfeur aléatoire.

Un autre algorithme proche de celui de Haveliwala [Hav02] à été proposé par Nie et al. [NDQ06]. L'idée de base de cet algorithme d'analyse de liens thématiques est de combiner le poids d'importance d'une page avec les thématiques abordées dans cette page. En effet, au lieu de tenir compte de la thématique connue des pages sources des liens entrants, l'algorithme calcule deux vecteurs associés à chaque page Web : le vecteur du contenu et le vecteur d'autorité. Le vecteur du contenu C_u : $[C(u_1), C(u_2), ..., C(u_T)]$ est une distribution de probabilité sur les différents thèmes abordés dans la page u. Ce vecteur représente le contenu de la page u dont chaque composant du vecteur représente la contribution relative de chaque thème par rapport au contenu global de la page. Le vecteur du contenu est statique et déterminé par l'analyse du contenu textuel de la page. En fait,

Nie et al [NDQ06] utilisent une classification de termes pour chaque document par rapport à un ensemble pré-défini de thèmes en vue de déterminer le vecteur de contenu du document. Comme le montre la figure 2.7, le contenu d'une page Web est représenté par un vecteur de contenu correspondant à chaque thème de l'ensemble pré-défini de thèmes. Ce vecteur est normalisé de telle sorte que la somme des probabilités vaut 1. De la même manière, une requête q est également considérée comme un document court associé à un vecteur de contenu de la requête C_q : $[C(q_1), C(q_2), ..., C(q_T)]$. Ce vecteur indique la pertinence de la requête q par rapport à chaque thème. En plus du vecteur du contenu, chaque page est associée à un autre vecteur dit "vecteur autorité" A_u : $[A(u_1), A(u_2), ..., A(u_T)]$ pour mesurer son importance. $A(u_k)$ représente le poids d'importance de la page u par rapport au thème k. Ce vecteur autorité est obtenu à partir d'un algorithme thématique qui propage dynamiquement des poids autorités à travers la structure hypertexte des liens. D'après la figure 2.7 la somme $A(u) = \sum_{k \in T} A(u_k)$ est identique au poids d'importance original de pagerank de la page u. Une fois le vecteur autorité d'un document D est calculé, un score de pertinence spécifique du document u par rapport à la requête q peut être calculé de la manière suivante :

$$S_q(u) = \sum_k A(u_k) * C(q_k) \tag{2.17}$$

où les éléments du vecteur autorité A_u sont pondérés en fonction de la distribution de la pertinence de la requête dans C_q.

Afin de calculer les vecteurs autorités des pages, [NDQ06] ont décrit un modèle de PageRank thématique reposant sur un surfeur aléatoire thématique. Un surfeur aléatoire thématique est similaire à un surfeur aléatoire décrit dans le modèle de PageRank. la seule différence est que le surfeur thématique est sensible aux différents thèmes de la page. Considérons un surfer aléatoire thématique parcourant des pages Web et supposons que le surfeur passe en revue un page Web v dont il est intéressé par le thème *k* dans la page *v*. Pour le prochain mouvement, le surfeur peut choisir soit de suivre un lien sortant de la page courante avec une probabilité $(1 - d)$, soit d'aller vers n'importe quelle page aléatoire en saisissant l'URL de la page avec une probabilité d (d=0.15). Intuitivement, en suivant un lien, le surfer est susceptible ou bien de rester sur le même thème avec une probabilité α pour maintenir la continuité du thème (followstay, "FS"), comme le montre la figure 2.7 ou bien d'aller à n'importe quel thème i dans la page cible du lien (follow-jump, "FJ") avec une probabilité $(1 - \alpha)$. En prenant un lien avec l'action "FJ", la préférence parmi les différents thèmes de la page cible est déterminé par le contenu de la page cible u, c-à-d., C_u. Dans l'exemple donné dans la figure 2.7, les probabilités "FJ" de v_1 à u_1, u_2 *et* u_3 sont $(1 - d) * (1 - \alpha) * C(u_1)$,$(1 - d) * (1 - \alpha) * C(u_2)$ et $(1 - d) * (1 - \alpha) * C(u_3)$ respectivement. Tandis que, la probabilité "FS" de v_1 à u_1 est de $(1 - d) * \alpha)$.

2.4.3.2 Utilisation des liens au niveau blocs physiques

Cai et al. [CYWM04] ont proposé deux algorithmes d'analyse de liens au niveau de blocs en considérant des unités plus petites que la page Web elle-même et ayant une pertinence sémantique. Ces unités appelées "*blocs sémantiques*" sont extraites en utilisant une méthode de segmentation de pages Web à critères visuels VIPS Cai et al. [CYWM03]. Les travaux de Cai et al. [CYWM04] repose sur l'hypothèse suivante : "*un document contient souvent différentes blocs (bloc publicitaire, bloc de navigation, bloc de contenu textuel, etc.) et des liens hypertextes contenu dans ces blocs et qui pointent vers des pages appartenant à différentes thématiques. L'existence de ces blocs peut conduire à des erreurs flagrantes si l'analyse des co-citations se fait au niveau des pages, en associant la même thématique à deux pages de contenu totalement différent*". En effet, un lien dans un bloc publicitaire n'est pas pertinent à la requête utilisateur.

La première tâche importante consiste à identifier les blocs sémantiques différents sur un document. Pour cela, Cai et al. [CYWM04] utilisent l'algorithme VIPS (VIsion-based Page Segmentation) : Segmentation du

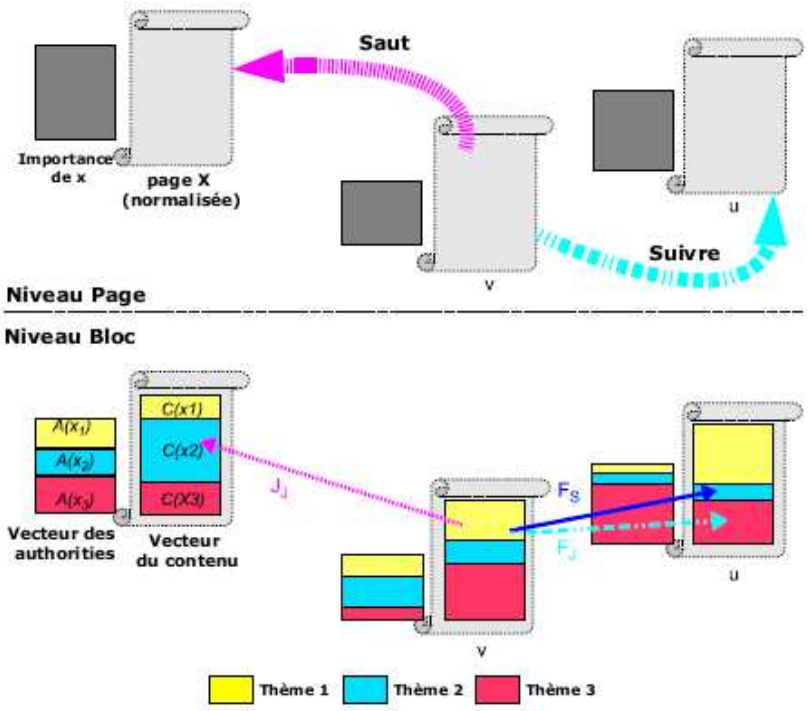


FIGURE 2.7: Les nœuds thématiques

document Web sur des critères visuels Cai et al. [CYWM03] . Il s'agit d'extraire la structure d'un document en utilisant des informations données par sa présentation visuelle (en fait les balises du code HTML du document). Le résultat est une structure arborescente, dans laquelle chaque nœud constitue un bloc. Les blocs se voient attribuer un "degré de cohérence", qui permet de déterminer à quel point le bloc en question est clairement séparé des autres blocs. Une solution possible est que les documents seront segmentés, par exemple, par des liens horizontaux et verticaux. Le contenu du bloc est pondéré en fonction de sa position dans le document. Les liens publicitaires comptent moins par rapport aux liens au milieu du contenu d'un bloc. En théorie, d'autres aspects visuels du document Web comme la couleur du fond, ligne et le font du texte peuvent être utilisés pour segmenter et pondérer les blocs.

L'algorithme proposé classe les documents Web par l'extraction des relations existantes entre document-bloc et bloc-document, puis l'utilisation de ces informations pour la construction d'un graphe de document et d'un graphe de blocs. Les relations document-bloc sont déterminées par l'analyse de la topographie de la page, et les relations bloc-document sont déterminées par la probabilité qu'un bloc pointe vers un document donné. Le but est de construire un graphe sémantique de telle sorte que chaque nœud représente exactement un seul bloc sémantique. Une fois l'importance donnée à un bloc est calculée, l'information serra utilisée dans les algorithmes d'analyse des liens :

- Block Level PageRank qui repose sur le calcul du PageRank (Brin et al. [BP98]).
- Block Level Hypertext-Induced Topic Selection (HITS) qui assignent une valeur d'importance à chaque document en se basant sur le type du bloc qui le pointe (bloc autorité ou bloc pivot) (Kleinberg [Kle99]).

A ce stade, il est possible d'éliminer le bruit, en identifiant les blocs de navigation, de publicité, et les éléments de décoration, qui sont en général facilement reconnaissables. La méthode a également permis aux chercheurs de calculer un BlockRank au niveau bloc semblable au PageRank au niveau page.

La comparaison entre les algorithmes d'analyse de liens au niveau bloc et les algorithmes standard comme pagerank Brin et al. [BP98] et HITS (Kleinberg [Kle99]) appliqués sur des collections standards en recherche

d'information montre que les algorithmes reposant sur l'analyse des blocs sont meilleurs que les algorithmes standard. Cependant, l'amélioration obtenue en termes de pertinence est marginale car comme d'habitude noyée dans les autres critères comme le choix de l'ensemble de test, les requêtes exécutées et les paramètre de l'algorithme de segmentation utilisés. De plus, il existe de nombreuses autres approches qui permettent d'améliorer la pertinence sans utiliser l'analyse des liens. L'analyse des liens au niveau des blocs du Web a pu également mener à une meilleure compréhension du graphe du Web en général. Cependant, la determination de ces blocs n'est pas facile à réaliser. La plupart des algorithmes de segmentation que nous avons cités dans ce chapitre considèrent quelques critères comme délimiteurs de segments alors qu'il existe une multitudes de critères de segmentation. Chaque auteur d'une page Web utilise le critère ou les critères qu'il trouve pertinent au découpage de sa page. De ce fait, plusieurs solutions sont à envisager selon les critères figurant dans la page. Le problème qui se pose maintenant est quels sont les délimiteurs de segments d'une page et comment les choisir ? Nous essayons de répondre à cette problématique dans la partie modèle de notre travail (chapitre 3).

2.5 Conclusion

Dans ce chapitre, nous avons défini les principes généraux qui régissent les modèles de propagation en recherche d'information et répertorié les algorithmes les plus classiques. Nous avons voulu nous démarquer des autres états de l'art qui existent sur le sujet par un éclairage différent, avec notamment une classification de ces modèles selon plusieurs critères :

- dépendance du système par rapport à la requête (systèmes indépendants Vs systèmes dépendant)
- type de propagation (popularité ou pertinence)
- paramètre de propagation (fixe ou dynamique)
- granularité d'information considérée (blocs thématiques, pages, groupe de pages)

Nous avons soulevé certains problèmes majeurs de chaque approche étudiée dans l'état de l'art. En effet, plusieurs travaux ont été menés sur l'utilisation des liens dans la recherche d'information sur le Web, mais jusqu'à maintenant de nombreuses expériences ont montré qu'il n'y a pas de gain significatif par rapport aux méthodes de recherche reposant seulement sur le contenu [SR00] [GCH+01b].

Les méthodes de propagation de popularité souffrent du problème de spamming des liens, de l'auto-citation, de l'avantage cumulé et non prise en compte de thèmes. Ces dernières introduisent du bruit dans le calcul des indices de popularité que nous avons étudiés dans ce chapitre (Indegree, PageRank, poids autorités et poids pivots). En effet, ces méthodes considèrent tous les liens hypertextes comme des liens de citation ou de référence. Or, ils existent des liens vides non porteurs d'informations additionnelles à la recherche demandée tels que les liens publicitaires, les liens de navigation et les liens artificiels qui ne doivent pas être pris en compte dans la propagation des indices de popularité.

Concernant la propagation de pertinence, nous venons de voir que le paramètre de propagation de pertinence est fixe et que les résultats obtenus avec ces modèles dépendent du paramètre de propagation utilisé et de l'ensemble des requêtes exécutées. Or, ces modèles de propagation de pertinence ne tiennent pas compte du nombre de termes de la requête utilisateur dans le calcul de la pertinence des documents par rapport à la requête utilisateur. Les techniques de propagation de pertinence sont moins vulnérables au spamming des liens par rapport aux techniques de propagation de popularité. Le fait d'accorder un score de pertinence à chaque nœud du sous graphe de la requête avant d'effectuer la propagation permet de donner plus d'importance au pages contenant toute l'information recherchée pa rapport aux pages contenant des informations partielles de la recherche effectuée. Par conséquent, les requêtes contenant un nombre faible de termes ne doivent pas être traitées de la même manière que les requêtes contenant un nombre important de termes. Plus la requête contient de termes, plus le nombre de pages de l'espace de travail de la requête est grand, plus le risque d'avoir beaucoup de liens non pertinent à la recherche serra grand (i.e., des liens qui relient les pages non pertinentes à la requêtes

mais contenu des informations partielles de la recherche (un nombre faible de termes de la requête)). Dans cette perspective, nous proposons dans ce travail de thèse un un modèle similaire aux modèles de propagation de pertinence. Cependant, au lieu de fixer le paramètre de propagation, nous allons le calculer dynamiquement pendant l'interrogation du système en fonction du nombre de termes de la requête utilisateur afin de favoriser les requêtes contenant un nombre important de termes par rapport aux requêtes contenant un nombre faible de termes. De plus, de pondéré les liens par rapport aux nombre de termes contenu dans la page source du lien afin de donner plus d'importance aux pages qui contiennent toute les informations désirées au détriment des pages qui contiennent des informations partielles de la recherche. Ce modèle serra détaillé dans le chapitre suivant.

Nous venons aussi de voir que les différentes techniques de propagation étudiées dans ce chapitre souffrent du problème du sens des thèmes. En effet, ces techniques considèrent la page Web comme l'unité d'information la plus petite à retourner à l'utilisateur. Or, la plupart des pages Web traitent plusieurs sujets de thématiques différentes et les liens hypertextes ne pointent pas la page cible entière mais la partie de la page de la même thématique que la page source du lien. De plus, il existe des parties d'une page Web qui ne sont pas pertinentes à la requête utilisateur et d'autres parties qui n'ont rien à voir avec les thématiques abordées dans la page et qui faussent le calcul de pertinence de la page par rapport à un besoin utilisateur. Ces parties correspondent aux barres de navigation, de publicité et d'offres commerciales et ne doivent pas intervenir dans le calcul de la pertinence d'une page. Ainsi, la détection des parties d'une page comme structures de contenus thématiques dans le but de calculer une certaine pertinence au niveau d'une unité inférieure que celle de la page devient un facteur potentiel pour l'amélioration des performances de la recherche d'information sur le Web. Le recours à la segmentation des pages est indispensable pour procéder au découpage de ces pages en blocs thématiques. Et comme les pages HTML ont une structure qui reflète son organisation en blocs, faire appel à des méthodes qui combinent les avantages des méthodes de segmentation du texte et d'autres méthodes reposant sur des critères de segmentation tels que les balise HTML s'avère très prometteur. En effet, il existe une multitudes de critères de segmentation mais le problème qui se pose est comment choisir ces critères afin de segmenter les pages Web. Dans la deuxième partie de notre modèle, nous allons proposer un algorithme de segmentation inspiré des algorithmes génétiques qui permet de choisir la meilleur solution de segmentation dans un espace de solutions générées à partir de différents critères de délimiteurs de segments. Dans ce qui suit, nous introduisons notre modèle de propagation dynamique ainsi que l'algorithme thématique de segmentation que nous avons proposée.

Chapitre 3

Modélisation du nouveau système

3.1 Introduction

Dans l'état de l'art, nous avons vu que les techniques existantes de propagation de popularité ou de pertinence en recherche d'information reposent sur l'importance du voisinage d'un document dans le calcul du score de pertinence du document par rapport à la requête utilisateur. Ces méthodes de propagation consistent à affiner la pertinence d'un document par rapport à la requête posée en fonction de la pertinence ou de l'indice de popularité des documents voisins. Cependant, les techniques de propagation étudiées dans l'état de l'Art ne font pas la distinction entre les documents qui contiennent toute l'information recherchée (les documents contenant tous les termes de la requête) et les documents qui contiennent des informations partielles à la recherche (documents contenant moins de termes que ceux de la requête). En effet, un lien qui part d'un document contenant tous les termes de la requête considéré comme un lien de qualité véhicule plus d'information qu'un lien qui part d'un document qui contient moins de termes de la requête. Ne pas tenir compte de la qualité des liens engendre du bruit dans le calcul de la pertinence des documents. A partir de cette observation, nous proposons une autre façon de calculer le score de pertinence d'un document par rapport à la requête utilisateur. L'idée principale de notre approche est de donner plus d'importance au voisinage des documents qui contiennent plus de termes de la requête par rapport au documents contenant moins de termes de la requête. L'hypothèse sur laquelle repose notre travail de recherche tient compte du nombre de termes distincts de la requête contenus dans les documents voisins dans le calcul d'un score de pertinence d'un document par rapport à la requête. Plus le voisinage du document contient un nombre important de termes de la requêtes, et plus le score de pertinence du document est grand.

Dans ce chapitre, nous définissons un nouveau modèle en recherche d'information reposant sur une fonction de correspondance qui tient compte, en plus du contenu informationnel des documents, du voisinage immédiat des documents jugés pertinents à la recherche effectuée. C'est un modèle de propagation de pertinence qui repose sur la pondération dynamique des liens hypertextes entre documents en fonction du nombre de termes de la requête contenus dans ces documents. Ce modèle s'appuie sur l'exhaustivité de la recherche. Notre hypothèse stipule que : ***"Les documents qui répondent totalement à la requête utilisateur sont plus important que les documents qui répondent partiellement à la requête utilisateur."***. Nous pondérons les liens hypertextes qui relient les documents d'une manière à favoriser les documents dont le voisinage contient plus de termes de la requête. Avec notre modèle, un document aura un score de pertinence important si :

1. Le voisinage du document est composé de documents contenant un nombre important de termes de la requête.
2. Le voisinage du document est composé de plusieurs documents répondant à la requête.

De plus, étant donné que la plupart des applications d'analyse de liens considèrent la page Web en tant que plus petite unité d'information indivisible, la pertinence du document par rapport à la requête utilisateur est calculée au niveau page. Or, une page est souvent composée de plusieurs contenus de thématiques différentes qui ne sont pas tous appropriées au contenu de la requête. Ainsi, une page ne devrait pas être la seule et la plus petite unité d'information sur le Web. Une page peut contenir plusieurs parties sémantiques dites blocs qui ne sont pas nécessairement harmonisées. Le calcul de la pertinence d'un document ne devrait pas se limiter à la page comme la seule granularité d'information sur le Web. En effet, une étude montre que dans la perception humaine, une page est perçue en tant que différents objets sémantiques plutôt qu'un seul objet (Bernard [Ber02]). Les utilisateur distinguent les différents blocs d'une page grâce aux indicateurs visuels contenu dans cette page tel que les couleurs, les lignes horizontales, les lignes verticales et la représentation structurelle de la page (paragraphes, titres). Par conséquent, la détection des structures de contenu thématique devient un facteur potentiel pour l'amélioration des performances de la recherche d'information sur le Web. Donc, il vaut mieux calculer la pertinence du document au niveau bloc thématique

Nous suggérons d'utiliser les critères visuels afin de déterminer les frontières entre les différents blocs d'une page. Le problème qui se pose maintenant est de savoir comment choisir ces critères visuels sachant qu'il y a une multitude de critères visuels existants et que chaque conception d'une page diffère de la conception d'une autre page dans le choix des délimiteurs de blocs. Nous avons opté pour l'analyse thématique des blocs résultants afin de choisir la meilleure segmentation et les meilleurs délimiteurs visuels de blocs dans une page. Notre travail consiste alors à extraire des blocs à partir des pages en utilisant la structure HTML de la page (les critères visuels et de présentation). Nous avons opté pour les algorithmes génétiques afin de mieux segmenter les pages Web. Nous allons détaillé cet algorithme dans les sections qui suivent. Cet algorithme repose sur le choix de la meilleure solution de segmentation parmi un ensemble de solutions de segmentation prévisibles de telle sorte que les contenu des blocs résultants soient cohérents et homogènes à l'intérieur de leur contenu et distants entre eux.

Une fois les blocs extraits, nous adaptons notre modèle de recherche d'information reposant sur une fonction de correspondance qui prenne en compte à la fois le contenu des blocs thématiques constituant la page et le voisinage de cette page constitué des blocs des pages citant cette page. Ce voisinage est calculé dynamiquement en pondérant les liens hypertextes reliant les blocs thématiques aux autres blocs en fonction des termes de la requête contenus dans ces blocs thématiques.

Afin de voir l'impact de notre modèle sur la recherche d'information, nous l'avons appliqué à des niveaux d'abstraction différents. Nous distinguons trois niveaux standards de granularité d'information (bloc, page et site) que les chercheurs utilisent pour affiner la pertinence d'un document par rapport à la requête utilisateur en ajoutant des informations additionnelles à la recherche effectuée.

Le niveau bloc représente l'unité d'information la plus petite et le niveau page l'unité d'information la plus connue et dont le résultat de la recherche n'est autre que les pages qui répondent à la requête utilisateur. Le dernier niveau est celui du site qui regroupe des pages étroitement liées à l'activité du site, soit des pages qui ont des informations en commun : par exemple le site d'une université regroupe des pages liées à l'enseignement universitaire. L'information sur le site des pages retournées est une information additionnelle qui renforce la pertinence des pages du site en calculant une valeur de pertinence du site sur son contenu global par rapport à la requête utilisateur. Le site qui contient un nombre important de pages en réponse à une requête utilisateur est considéré comme un site spécifique à la requête utilisateur. D'où un score important assigné aux pages de ce site.

Dans ce qui suit, nous présentons en premier le modèle générique de propagation de pertinence que nous proposons. Puis, nous décrivons l'architecture du système à plusieurs niveaux d'abstraction (bloc, page et site) ainsi que la formalisation mathématique de chaque niveau d'abstraction. Ensuite, nous présentons l'algorithme

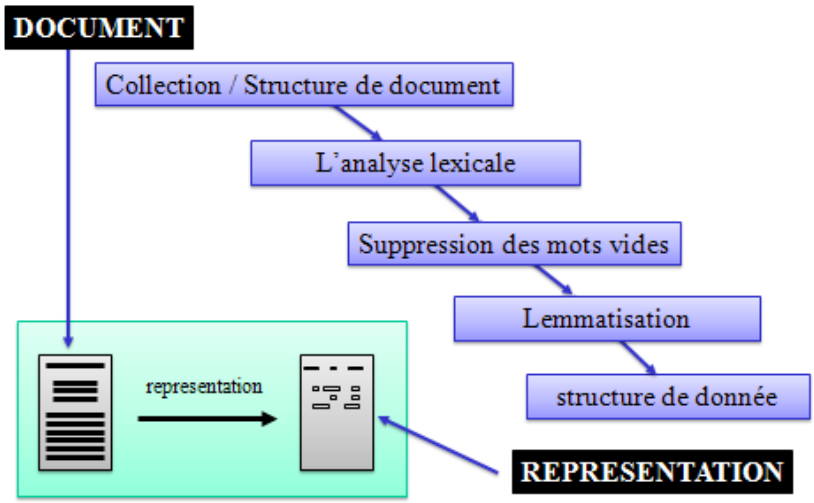


FIGURE 3.1: Les étapes d'indexation

de segmentation des pages Web qui permet de découper les pages Web en plusieurs blocs de thématiques différentes. Nous allons montrer les différentes étapes de cet algorithme et le processus de segmentation simplifié que nous avons conçu.

3.2 Modèle de propagation de pertinence

Par définition, un modèle en recherche d'information est composé d'un module d'indexation qui représente les documents, d'un module d'interrogation qui représente les requêtes et d'une fonction de correspondance qui calcule un degré d'appariement entre les termes d'indexation des documents et les termes de la requête. Dans ce qui suit nous décrivons les étapes d'indexation des documents et la représentation du contenu de ces documents. Puis, nous proposons une nouvelle technique de pondération des termes de l'index. Enfin, nous détaillons la fonction de correspondance que nous proposons ainsi que les différentes variantes de notre fonction de correspondance.

3.2.1 Représentation des documents

Cette section montre comment les documents sont transformés en vecteur de poids de termes pour être utilisés par notre approche de recherche. En général, les représentations n'utilisent pas d'information grammaticale ni d'analyse syntaxique des termes, seule la présence ou l'absence des termes est porteuse d'informations. La représentation des documents que nous avons adopté a été introduite pour la première fois dans le cadre du modèle vectoriel (Salton et al. [SWY75]). Les documents sont transformés simplement en vecteurs dont la i^{me} composante représente le poids du i^{me} terme. La premiere étape de la transformation des documents en vecteurs de poids de termes consiste à extraire tous les termes d'un document. Dans les langues comme le français ou l'anglais, les termes sont séparés par des espaces ou des signes de ponctuations. Cependant, il existe plusieurs formes d'un terme qui sont considérées comme des termes différents, alors qu'il ne s'agit que de formes dérivées de la même racine du terme et qui ont a priori le même sens (exemple bank, banking, banks, etc). Pour remédier à ce problème, il est possible de considérer uniquement la racine des termes plutôt que les termes entiers (on parle de "stem" en anglais). Plusieurs algorithmes ont été proposés pour substituer les termes par leur racine lexicale, l'un des plus connus pour la langue anglaise est l'algorithme de Porter [Por80] que

nous avons utilisé sur nos corpus de documents (WT10g et Gov de TREC). La substitution des termes par leur racine réduit considérablement l'espace des termes et permet de représenter par un même terme des termes qui ont le même sens. Par exemple, le remplacement des termes *performs*, *performed*, *performing* par l'unique racine *perform* semble être avantageux. Néanmoins ces substitutions peuvent augmenter l'ambiguïté des termes en représentant par un même terme des termes de sens différents (exemple action(agir), action(bourse)). La distribution des termes dans un corpus de documents n'est pas uniforme. Certains termes apparaissent très fréquemment, tandis que d'autres n'apparaissent que très rarement. Comme le nombre de termes présents dans un corpus peut être très grand, les méthodes statistiques cherchent, en général, à réduire le nombre de termes utilisés pour représenter les documents. Les termes qui apparaissent le plus souvent dans un corpus de documents sont les déterminants, les pronoms et mots de liaison. Ces termes doivent être supprimés de la représentation des documents pour deux raisons :

- D'un point de vue linguistique, ces termes ne contiennent que très peu d'informations. La présence ou l'absence de ces termes n'aident pas à deviner la portée d'un document. Pour cette raison, ils sont souvent appelés *mots vides* (ou *"stop words"* en anglais).
- D'un point de vue statistique, ces termes se retrouvent dans l'ensemble des documents d'une collection sans aucune discrimination et ne sont d'aucune aide pour la recherche d'information pertinente.

Comme le nombre de termes les plus fréquents est faible, il est souhaitable de définir une liste de termes qui seront automatiquement supprimés de la représentation d'un document. Cependant, l'établissement d'une telle liste peut poser des problèmes. D'une part, il n'est pas facile de déterminer le nombre exacts de termes qu'il faut inclure dans cette liste. D'autre part, cette liste est étroitement liée à la langue utilisée et n'est donc pas transposable directement à d'autres langues. Dans notre modèle, nous avons utilisé une liste contenant 956 termes jugés par des professionnels comme étant des termes vides. En plus des mots vides, les méthodes statistiques cherchent à supprimer les mots rares d'un corpus de documents afin de réduire d'une façon considérable la dimension des vecteurs utilisés pour représenter ces documents. En effet, il existe un grand nombre de termes très rares n'apparaissant qu'une fois ou deux dans le corpus et il existe tout un ensemble de termes dont la fréquence d'apparition est faible dans le corpus du document. D'un point de vue linguistique, la suppression de ces termes n'est pas nécessairement justifiée, certains termes peuvent être très rares, mais très informatifs. La suppression de ces termes peut engendrer une perte d'information du document. C'est pour cette raison que nous avons décidé de les garder car il n'est pas possible de savoir a priori quels sont les termes non informatifs parmi l'ensemble de termes rares. Les dimensions des vecteurs des deux collections de TREC $WT10_g$ et GOV sont respectivement de l'ordre de $2,222,214$ termes et $2,046,594$ termes dont la moitié ($1,103,603$ termes pour $WT10_g$ et $1,133,515$ pour Gov) figurent une seule fois dans la collection entière. La figure 3.1 montre le processus d'indexation de chaque document de la collection.

3.2.2 Représentation des requêtes

La représentation des requêtes est la même que celle des documents où chaque requête est représentée par un vecteur de termes. Nous distinguons deux types de requêtes : les requêtes de selection et les requêtes de description. Une requête de selection contient un nombre faible de termes et sert uniquement à retrouver les documents en réponse à la requête posée par l'utilisateur. Tandis qu'une requête de description contient un nombre important de termes. En effet, la requête descriptive contient les termes de la requête de selection, les synonymes de ces termes ou les termes qui décrivent la recherche attendue du système. Les requêtes descriptives peuvent être construites de deux façons différentes : d'une part, l'utilisateur peut saisir en plus des termes de la requête de selection, d'autres termes susceptibles d'être proches de la recherche demandée. Ces derniers n'interviennent pas dans le processus de selection des documents en réponse à la requête utilisateur. Une autre façon de construction de la requête descriptive consiste à faire appel à des mesures de similarité entre termes. Cette dernière permet de créer les requêtes descriptives automatiquement en raffinant la requête de selection

par des termes similaires à ceux de la requête de selection. Lors de l'interrogation du système, la recherche s'effectue en deux étapes. La premiere étape consiste à retrouver les documents selon l'existence des termes de la requête de selection dans ces documents. La deuxième étape consiste à calculer les scores de pertinence des documents sélectionnés en fonction des termes de la requête descriptive. Dans les deux collections de test WT10g et GOV que nous avons utilisées dans nos expérimentations, l'ensemble des requêtes exécutées sont composées de deux champs : Le titre et la description de la requête. Le premier champ correspond à nos requêtes de sélection et le deuxième champ à nos requêtes de description. Nous avons aussi ajouter aux requêtes de description des termes similaire aux termes de la requête de sélection en utilisant la co-occurrence des termes. Nous gardons que les termes dont la similarité entre ces termes et ceux de la requête de sélection est supérieure à un seuil fixe. Nous ne prenons que les cinq meilleurs termes en terme de similarité de co-occurrence.

3.2.3 Indexation

Considérons maintenant la pondération des termes d'indexation d'un document. Alors qu'en recherche d'information traditionnelle, le poids d'un terme dans un document combine d'une part une importance locale de ce terme à l'intérieur du document et d'autre part une importance globale au sein de toute la collection. En effet, l'importance locale d'un terme peut être calculée en fonction du nombre d'occurrences du terme dans le document, tandis que l'importance globale peut être calculée en fonction du nombre d'occurrences du terme dans toute la collection.

Dans notre travail de recherche, nous introduisons un nouveau facteur d'importance du terme dans un document. Ce facteur est calculé en fonction du voisinage immédiat du document qui contient le terme. Cette importance vue de l'extérieur du document permet de renforcer les poids des termes du texte ancre des liens entrants et ceux autour du texte ancre des liens. En effet ces termes sont considérés comme des descriptifs de la page cible du lien.

De nombreuses solutions ont été proposées dans la littérature pour prendre en compte les termes du texte ancre des liens dans le calcul de pondération des termes en assignant d'une part ces termes à la page cible du lien et d'autre part de donner plus d'importance aux poids de ces termes en les multipliant par un facteur fixe (double, triple, etc.). Cependant, le calcul du poids de ces termes ne tient pas compte de l'importance de ces termes dans le voisinage du document. En effet, il existe plusieurs types de liens tels que les liens de navigation, liens de citations et liens de structure. Par exemple, dans un lien de navigation, les termes utilisés dans le texte ancre de ces liens n'ont rien avoir avec le contenu informationnel du document, par conséquent, ces termes ne doivent pas avoir plus d'importance. Tandis que dans un lien informationnel dont le contenu du document source et du document destinataire du lien sont complémentaires ou étroitement liés, associer une grande valeur d'importance à ces termes peut s'avérer utile pour améliorer les performances de la recherche. C'est pour cette raison que nous proposons une nouvelle manière de calculer cette importance en se basant sur notre modèle générique de propagation de pertinence. Nous distinguons deux facteurs d'importance du terme par rapport à un document :

– Le premier facteur d'importance est la visibilité d'un terme dans un document par rapport à son voisinage immédiat. Ce facteur, dit facteur de visibilité d'un terme et noté $V_T(T, D)$, représente la fraction du nombre de documents qui pointent vers *D* et qui contiennent le terme *T* par rapport au nombre de documents qui pointent vers *D*. Ce facteur est défini comme suit :

$$V_T\,(T, D) = \frac{|V_D(T)|}{|V_D|} \tag{3.1}$$

Où V_D représente l'ensemble des documents qui pointent vers *D* (voisinage de *D*) et $V_D(T)$ l'ensemble du voisinage de document *D* contenant le terme *T*.

Ce facteur de visibilité nous permet de connaître le pourcentage de représentation du terme *T* dans le voisinage de *D*.

– Le deuxième facteur n'est autre que le poids du terme *T* calculé à partir du voisinage du document en tenant compte de sa visibilité. Ce poids est calculé de la manière suivante :

$$PoidsV_T\,(T,D) = \sum\nolimits_{D_i \in V_D(T)} \frac{V_T\,(T,D)}{|V_D(T)|} * Okapi(T,D_i) \tag{3.2}$$

Nous avons opté pour une pondération Okapi BM-25 afin de calculer le poids d'un terme *T* dans un document *D*. Cette pondération tient compte du nombre d'occurrences du terme *T* dans le document *D* : $TF(T,D)$, du nombre d'occurrences du terme *T* dans toute la collection $QTF(T,C)$ et de la taille du document *dl(D)*. Cette mesure est calculé comme suit :

$$Okapi_T(T,D) = W_{rs}(T)\frac{(K1+1)*TF(T,D)}{K+TF(T,D)} * \frac{(K3+1)*QTF(T,C)}{K3+QTF(T,C)}) \tag{3.3}$$

avec $W_{rs}(T)$ est le poids de Robertson/Sparck Jones du terme *T*. Il est calculé comme suit :

$$W_{rs}(T) = log(\frac{N-n+0.5}{n+0.5})$$

Où N représente le nombre de documents dans la collection, n le nombre de documents contenant le terme *T* dans la collection. *K* est calculé comme suit :

$$K = K1 * ((1-b) + b * (dl(D)/avdl))$$

Où *dl(D)* et *avdl* représentent respectivement la taille en octets du document *D* et la taille moyenne en octets d'un document dans toute la collection.

Dans nos expériences, nous fixons K1 = 4.2, K3 = 1000, b = 0.8. Ces valeurs ont permis de réaliser de meilleurs résultats avec l'algorithme de base reposant sur le contenu seul des documents dans la conférence TREC de 2000.

Le poids final d'un terme *T* dans un document *D* est une combinaison des deux poids (Okapi et poids du voisinage). Soit $Poids_T(T,D)$ le poids final du terme T dans un document D. Ce poids est calculé comme suit :

$$Poids_T(T,D) = Okapi_T(T,D) + PoidsV_T\,(T,D) \tag{3.4}$$

3.2.4 Fonction de correspondance

Après avoir décrit le module d'indexation et le module d'interrogation, nous passons à la description de la fonction de correspondance que nous proposons. Cette fonction diffère des autres fonctions existantes dans la littérature par le biais de paramètre de propagation dynamique calculé en fonction du nombre du termes de la requête contenus dans les documents. En effet, toutes les techniques de propagation de pertinence ou de popularité existantes utilisent des paramètres statiques de propagation. C'est à dire la portion du score propagée d'un document à un autre est fixé par l'algorithme. Par conséquent, on a besoin de recalculer ces paramètres statiques en passant d'une collection ou d'une requête à une autre. C'est pour cette raison que nous nous sommes intéressé par le calcul dynamique du score de voisinage d'un document.

La fonction de correspondance de notre modèle dépend du contenu textuel des documents et de leurs voisinages. Cette dépendance permet une meilleure adéquation des résultats retrouvés par un modèle classique de

recherche d'information vis-à-vis d'un besoin utilisateur. Notre fonction de correspondance repose sur deux mesures : l'une est classique et utilisée dans les systèmes actuels. C'est la mesure OKAPI BM25 [RWB+92] et l'autre repose sur le calcul d'un score de voisinage dynamiquement en pondérant les liens hypertextes reliant les documents en fonction du nombre de termes de la requête contenus dans ces documents. La combinaison de ces deux scores est décrite dans la fonction suivante :

$$S_D(D,Q) = \alpha * Okapi_D(D,Q) + (1-\alpha) * V_D(D,Q) \tag{3.5}$$

Avec α un paramètre compris entre 0 et 1. Il nous permet de voir l'impact de notre fonction de voisinage sur celle reposant sur le contenu seul du document. On note $S_D(D,Q)$ le score de pertinence du document *D*, $V_D(D,Q)$ le score de voisinage du document *D* et $Okapi_D(D,Q)$ le score associé au document *D* reposant sur le contenu textuel du document par rapport à la requête. Nous détaillons la fonction de contenu OKAPI M25 et la fonction de voisinage dans la section suivante.

3.2.4.1 Fonction de contenu

Nous avons opté pour la formule d'OKAPI afin de calculer un score de pertinence reposant sur le contenu d'un document par rapport à une requête utilisateur. La formule d'OKAPI proposé par [RWB+92] et appliqué dans la recherche d'information est décrite comme suit :

$$\mathrm{Okapi_D(D,Q)} = \sum\nolimits_{T \in Q} \mathrm{Poids_T}(T,D) \tag{3.6}$$

Avec Q représente la requête utilisateur et $Okapi_D(D)$ le score de pertinence du contenu du document D par rapport à la requête utilisateur Q.

3.2.4.2 Fonction de voisinage

La fonction de voisinage que nous avons proposée tient compte de la structure du Web composée de liens hypertextes. L'hypothèse que repose notre modèle est décrite comme suit : ***on considère qu'un document D est bien connu pour un terme T de la requête Q si celui-ci contient beaucoup de liens entrants émis à partir des documents qui eux aussi contiennent le terme T de la requête (Doan et al. [DC05])***. Cette mesure tient compte du nombre de termes de la requête contenus dans les documents voisins. L'idée principale de notre mesure de voisinage est de pondérer les liens entrants selon le nombre de termes de la requête contenus dans les documents sources des liens. L'hypothèse stipule que le poids d'un lien émis par un document contenant *n* termes de la requête est 2 fois plus important que le poids d'un lien émis par un document contenant $n-1$ termes de la requête, 2^2 fois plus important que le poids d'un lien émis par un document contenant $n-2$ termes de la requête, ..., 2^{n-1} fois plus important que le poids d'un lien émis par un document contenant un seul terme de la requête. En effet, la mesure de voisinage est décrite comme suit :

$$V_D(D,Q) = \sum_{D_i \to D} \frac{Poids(D_i,D,Q) * Okapi_D(D_i,Q)}{|V_D|} \tag{3.7}$$

Avec V_D est l'ensemble des documents qui pointent le document *D* et $Okapi_D(D_i,Q)$ le score de pertinence reposant sur le contenu seul du document D_i par rapport à la requête *Q*. $Poids(D_i,D,Q)$ représente la pondération du lien entre le document D_i et le document *D* en fonction du nombre de termes de la requête *Q* contenus dans le document D_i. Plus le document D_i contient de termes de la requête *Q*, plus le poids du lien

entre D_i et *D* est grand et plus le score de pertinence du document *D* par rapport à la requête *Q* est important. Ce poids est défini comme suit :

$$Poids(D_i, D, Q) = \frac{2^k}{2^{ntq}} * \beta \tag{3.8}$$

Avec ntq représente le nombre de termes de la requête *Q* et *k* le nombres de termes de la requête *Q* contenus dans le document *D*. β est un paramètre compris entre 0 et 1 qui vérifie que la distribution des poids sur les différents types de liens soit une distribution de probabilité. Nous distinguons ntq types de liens par requête : les liens partant de documents contenant un seul terme de la requête, les liens partant de documents contenant deux termes de la requête, .., les liens partant de documents contenant tous les termes de la requête (soit *ntq*). La valeur de β est calculée pour chaque requête comme suit :

$$\sum_{k=1}^{ntq} \frac{2^k}{2^{ntq}} * \beta = 1 \quad \Rightarrow \quad \beta = \frac{1}{2*\left(1-\left(\frac{1}{2}\right)^{ntq}\right)} \tag{3.9}$$

Le paramètre β qui dépend essentiellement du nombre de termes de la requête permet de modérer l'impact du voisinage d'un document dans la fonction de correspondance. En effet, lorsque le nombre de termes de la requête est élevé, la taille de l'espace de travail (c.à.d l'ensemble des documents qui répondent à la requête) ainsi que le voisinage des documents dans cet espace de travail est grande, tandis que le paramètre β est moins important. Ce qui veut dire que l'importance accordée au voisinage des documents diminue en fonction de l'augmentation du nombre de termes de la requête. Plus le nombre de termes de la requête est grand, plus la valeur du paramètre β est petite et plus l'apport du voisinage des documents dans la fonction de correspondance est faible.

3.3 Architecture du système en trois couches

Dans la suite, nous utilisons les notations suivantes :

- Un graphe G représentant le Web est un couple (V, E), où V est l'ensemble des nœuds qui représente l'ensemble de blocs, l'ensemble de pages ou l'ensemble de sites Web et E est une partie de $V \times V$. On distingue deux types de graphes : les graphes orientés pour lesquels les éléments de E sont des couples (ordonnés) de nœuds et les graphes non orientés pour lesquels les éléments de E sont des paires (non ordonnées) de nœuds. Les éléments de E sont appelés des arcs dans le cas du graphe orienté et arêtes dans la cas du graphe non orienté.
- Un graphe valué est un triplet (V, E, ω) où $\omega : E \rightarrow \mathbb{R}$ est une fonction qui associe à chaque lien un poids réel positif. Ce poids désigne l'importance du liens qui relient deux nœuds du graphe.
- Le sous graphe $G' = (V', E')$ de G est un graphe dont l'ensemble des nœuds V' est un sous-ensemble de V et dont les liens de E' sont des liens de E qui relient les nœuds de V' dans G. Pour chaque nœud u du graphe G', un poids positif , noté $S(u)$, est associé à ce nœud. Ce poids désigne le score de pertinence du nœud par rapport à la requête utilisateur. Ce score est calculé en fonction du contenu textuel des nœuds par rapport à la requête utilisateur.
- Dans un graphe non orienté (resp. orienté), le degré d'un nœud u, noté $D(u)$, représente le nombre du nœuds qui y sont reliés à u. $D(u)$ représente le nombre de voisins immédiats du nœud u. Dans le cas d'un graphe orienté, on parlera de degré entrant, noté $D^+(u)$, qui est le nombre de liens entrants du nœud u et de degré sortant, noté $D^-(u)$, qui est le nombre de liens sortant du nœud u. On notera $N(u)$ l'ensemble des nœuds voisins de u, de même $N^+(u)$ l'ensemble des nœuds voisins citant le nœud u et $N^-(u)$ l'ensemble de nœuds voisins cités par le nœud u.

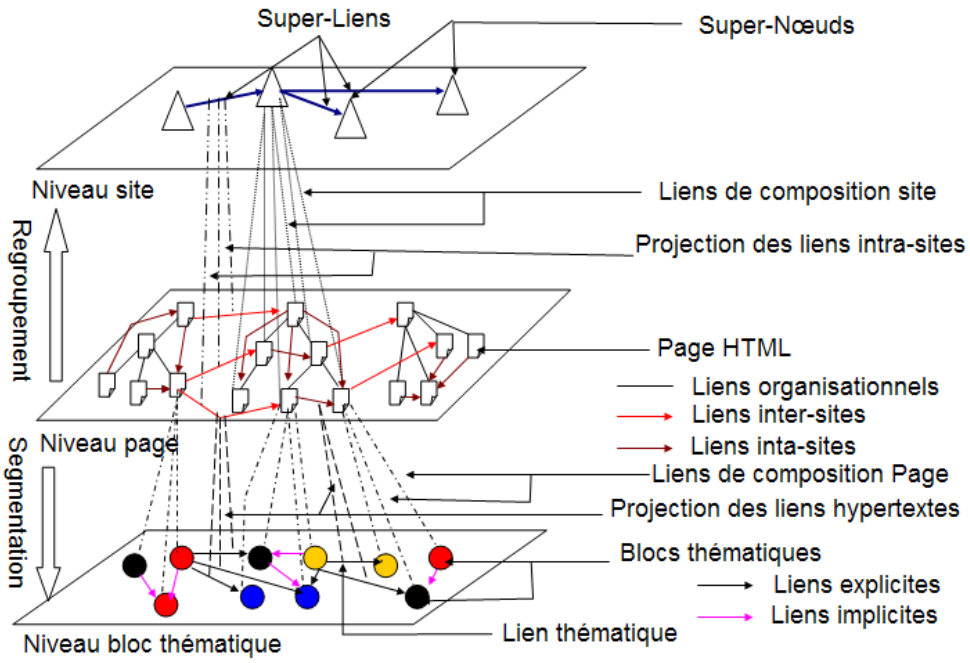


FIGURE 3.2: Architecture du système à trois niveaux

L'architecture du système sur laquelle repose notre modèle de recherche d'information est composée de trois niveaux de pertinence d'un document par rapport à une requête utilisateur : niveau bloc, niveau page et niveau site. En effet, lorsqu'un utilisateur recherche une information précise en saisissant des requêtes spécifiques, il est préférable de calculer la pertinence du document au niveau bloc et lorsqu'il recherche des informations imprécises en saisissant des requêtes génériques, il vaux mieux calculer la pertinence du document au niveau page. La différence entre une requête spécifique et une requête générique réside dans le choix des termes de la requête. Les termes génériques se trouvent en général dans les titres des pages Web, les textes ancres des liens entrants, ou au début du contenu textuel de la page. Ces termes dits génériques décrivent le contenu global de la page. Tandis que les termes spécifiques se trouvent dans le contenu textuel de la page, en particulier, dans des paragraphes au milieu du contenu du document. Ces termes dits spécifiques permettent d'avoir plus de détails sur le contenu global du document et des fois des informations additionnelles à la description globale du document.

Par exemple, rechercher une page personnelle d'un acteur ou d'un scientifique revient à saisir une requête générique avec des informations sur l'identité de la personne recherchée (termes génériques). Le contenu des pages retrouvées contient des informations additionnelles à la recherche telles que les activités de la personne concernée, ces nouvelles, etc (termes spécifiques). En effet, l'utilisateur s'intéresse à ces informations additionnelles en plus de l'information recherchée. Dans ce cas, calculer une pertinence au niveau page est le plus adapté à cette recherche (requête générique).

Alors que dans une recherche spécifique, le but est de retrouver l'information désirée qu'on voulait retrouvée. Par exemple, une recherche sur un événement précis qui a eu lieu dans le passé revient à saisir une requête spécifique avec des informations précises sur l'événement (date, lieu, etc). C'est à dire, il faut que la page retrouvée contient le bloc ou les blocs qui parlaient de cet événement. Dans ce cas de figure, calculer la pertinence des documents au niveau bloc est la plus adaptés.

Or, il est difficile de distinguer les requêtes génériques des requêtes spécifiques. En effet, étant donné que les besoins d'informations diffèrent d'une personne à une autre, une requête peut être vue comme spécifique

et générique en même temps par deux personnes différentes car il est difficile de traduire le besoin utilisateur en un ensemble de termes. Ce qui nous amène à calculer deux scores : un score générique en considérant la page Web comme étant l'unité d'information renvoyée par le moteur de recherche et un score spécifique en considérant les blocs thématiques. Enfin, lorsqu'un site contient un nombre important de pages en réponse à une requête utilisateur, on le considère comme un site spécifique à la requête utilisateur et les pages qu'il contient doivent avoir un score de pertinence important. Le score de pertinence final d'un document Web (dans notre cas la page Web) sera une combinaison des trois scores.

Pour toutes ces raisons, il est indispensable de calculer un indice de pertinence d'un document Web en fonction de différentes granularités d'information et de renvoyer à l'utilisateur des pages Web susceptibles de contenir des informations précises sur la recherche effectuée en donnant plus d'importance à des pages issues de sites pertinents vis-à-vis à la requête utilisateur. Afin de calculer cette pertinence, nous avons adapté le modèle générique de propagation de pertinence que nous avons proposé à différent niveaux de granularité d'information. Le but est d'enrichir le contenu d'un document Web à travers la structure des liens et de combiner des scores de pertinence de différentes granularités d'information. Nous allons définir pour chaque niveau de granularité d'information le graphe sur lequel repose le modèle et les formules de calcul de la pertinence. Dans ce qui suit nous détaillons les trois niveaux d'abstraction de pertinence des documents. Nous combinons les scores obtenus pour chaque niveau d'abstraction afin de voir l'impact de chaque mesure. Des expérimentations ont été réalisées dans ce but.

3.3.1 Niveau page

La page Web est l'unité standard de diffusion d'information sur le Web. Même si les pages Web contiennent des informations sur les différents thèmes abordés, elles traduisent d'une certaine façon les différents points de vue de l'auteur ou des auteurs de la page sur les différents thèmes traités dans la page. Le fait de renvoyer la page à un utilisateur en réponse à sa requête peut être vu comme une source d'informations complémentaires à la recherche effectuée qui peut servir à l'acquisition de nouvelles connaissances et aider l'utilisateur à affiner sa requête. Le graphe associé au niveau page n'est autre que le graphe du Web. Il est constitué de l'ensemble de pages de la collection ainsi que l'ensemble de liens hypertextes qui relient ces pages. La collection de documents peut être modélisée par un graphe de pages. C'est la modélisation la plus connue dans la littérature. Le graphe des pages $G_p(C, L_p)$ est défini par l'ensemble des pages de la collection de documents et l'ensemble des liens hypertextes qui relient ces pages. Les caractéristiques du graphe G_p sont :

1. C représente l'ensemble des pages de la collection. Dans notre cas, les collections WT10g et .Gov de TREC. Chaque nœud du graphe est une page Web.
2. L_p est l'ensemble de couples ordonnés(P_i,P_j) formés de deux pages de l'ensemble C qui matérialise l'existance d'un lien hypertexte entre les deux pages P_i et P_j. Afin de représenter le graphe mathématiquement, une matrice d'adjacence MP est associée au graphe. Chaque élément de la matrice $MP_{i,j}$ peut prendre comme valeur 1 ou 0. Elle est définie comme suit :

$$MP_{i,j} = \begin{cases} 1 \; S'il \; existe \; un \; lien \; entre \; la \; page \; p_i \; et \; p_j \\ 0 \; Sinon \end{cases}$$

Pendant l'interrogation du système, un sous graphe $G'_p = (C', L'_p, \omega_p)$ est associé à la requête utilisateur Q. Ce graphe dépend des termes de la requête Q. Il est caractérisé par l'ensemble des propriétés suivantes :

1. C' représente l'ensemble des pages de la collection C qui répondent à la requête utilisateur Q. Ces pages contiennent au moins un terme de la requête Q. Un score de pertinence reposant sur le contenu seul de la page, calculé par la formule OKAPI BM25, est associé à chaque page de C'. Soit $Okapi_D(P, Q)$ le score de pertinence de la page P par rapport à la requête utilisateur Q.

2. L'_p est l'ensemble de couples ordonnés (P_i, P_j) formés de deux pages dont l'une d'elle appartient à l'ensemble C'. Le fait de garder les liens qui partent de l'ensemble C' vers $C - C'$ ou les liens entrant de l'ensemble $C - C'$ vers C nous permet de distinguer d'une part les pages fortement pertinentes à la requête utilisateur Q et d'autre part les pages les moins pertinentes à la requête utilisateur ou les pages de type spam. En effet, une page est fortement pertinente à la requête utilisateur Q, si et seulement si, la page est reliée par des liens hypertextes à d'autres pages pertinentes elles aussi à la requête Q, c'est à dire des pages appartenant à l'ensemble C'. Tandis qu'une page est moins pertinente à la requête utilisateur Q ou une page est de type spam, si est seulement si, la page est reliée à d'autres pages qui ne satisfont pas à la requête utilisateur Q, c'est à dire des pages appartenant à l'ensemble $C - C'$.
3. ω_p est la fonction de pondération des liens qui associe à chaque lien de L'_p un poids calculé en tenant compte du nombre de termes de la requête Q contenu dans la page source du lien. Cette fonction est définie comme suit :

$$\begin{aligned} &\omega_{\mathrm{p}} : L'_p \rightarrow R \\ &(P_i, P_j) \rightarrow \omega_p\left(P_i, P_j, Q\right) = \frac{2^k}{2^{ntq+1} * \left(1 - \left(\frac{1}{2}\right)^{ntq}\right)} \end{aligned}$$

   Avec ntq et k représentent le nombre de termes de la requête utilisateur Q et le nombre de termes de la requête Q contenu dans la page P_i respectivement.

Une fois les scores de pertinence des pages Web et les poids des liens hypertextes sont calculés, nous appliquons notre modèle générique de propagation de pertinence pour propager les scores de pertinence des pages Web à travers les liens hypertextes du graphe de pages généré G'_p. Le voisinage d'une page est calculé comme suit :

$$V_p\left(P, Q\right) = \sum_{P_i \rightarrow P} \frac{\omega_p\left(P_i, P, Q\right) * Okapi_D\left(P_i, Q\right)}{|V\left(P\right)|} \tag{3.10}$$

Avec $V(P)$ l'ensemble des pages qui pointent la page P. $Okapi_D(P_i, Q)$ représente le score de pertinence reposant sur le contenu seul de la page P_i par rapport à la requête utilisateur Q. $\omega_p(P_i, P, Q)$ est le poids du lien hypertexte qui relie la page P_i à la page P_j.

Le score final de la pertinence d'une page P par rapport à la requête utilisateur Q est défini comme suit :

$$S_p\left(P, Q\right) = \alpha * Okapi_D\left(P, Q\right) + (1 - \alpha) * V_p\left(P, Q\right) \tag{3.11}$$

Avec α un paramètre compris entre 0 et 1. Ce paramètre nous permet de voir l'impact du voisinage de la page sur la pertinence de la page basée sur le contenu seul. $Okapi_D(P, Q)$ représente le score de pertinence reposant sur le contenu seul de la page P par rapport à la requête utilisateur Q. $V_p(P, Q)$ le voisinage de la page P par rapport à la requête utilisateur Q calculé en pondérant les liens hypertextes qui relient ces pages en fonction du nombre de termes de la requête contenu dans les pages sources des liens.

3.3.2 Niveau site

Le Web peut être vu comme une forêt où chaque arbre de la forêt représente une structure arborescente d'un répertoire, d'un site Web ou d'un nom du domaine dont chaque feuille de l'arbre représente une page Web. Ces arbres sont reliés par des super liens. Dans notre modèle, nous concéderons que les sites Web pour deux raisons :

1. Nous supposons qu'un site Web est créé par une seule personne ou un groupe de personnes travaillant ensemble dans un même domaine d'activité (entreprise,collectivité, université,..). Donc, la structure hiérarchique du site reflète réellement son organisation.

2. Dans la plupart des cas, un site Web contient une page d'accueil appelée index qui facilite la construction de l'arborescence du site à partir de cet index.

Nous supposons que les pages d'un site Web véhiculent deux types d'information :

1. Des informations communes en relation étroite avec la thématique et l'activité du site. Ces informations sont de grande utilité dans le calcul de la pertinence des pages du site par rapport à la requête utilisateur.
2. Des informations spécifiques à chaque page du site.

Un site Web contient des informations sur plusieurs pages. La qualité d'un site par rapport à un besoin utilisateur peut être vue comme un indice de pertinence des pages du site par rapport à la requête utilisateur. En effet, la qualité d'un site, calculée en tenant compte des scores de pertinence des pages du site répondant à la requête utilisateur et de son voisinage par rapport à d'autres sites, constitue un indice de pertinence du site par rapport à la thématique de la requête utilisateur. Le graphe associé au niveau site est une projection du graphe de pages sur le niveau site. Il est composé d'un ensemble de super nœuds correspondant aux sites Web ainsi qu'un ensemble de super liens correspondant aux liens entre sites. Un super lien représente la projection des liens hypertextes du niveau page sur le niveau site. L'ensemble des liens qui relient des pages de deux sites différents est traduit par un seul super lien qui relie ces deux sites. Le poids d'un super lien est calculé en fonction des poids de tous les liens hypertextes qui relient les pages des deux sites. Ce poids n'est autre que la somme des poids des liens hypertextes qui relient les pages des deux sites. Le score de pertinence d'un site est calculé en fonction des scores de pertinence des pages du site qui satisfont la requête utilisateur et de la profondeur de ces pages dans l'arborescence du site. Plus la profondeur d'une page est grande, moins la pertinence du site est élevée. En effet, dans nos calculs, nous favorisons les pages qui se trouvent proche de la racine du site car ces pages contiennent des informations génériques sur l'activité site, alors que les pages feuilles du site contiennent des informations spécifiques. L'analyse de l'URL d'une page nous permet d'extraire le nom du site et la profondeur de la page dans ce site.

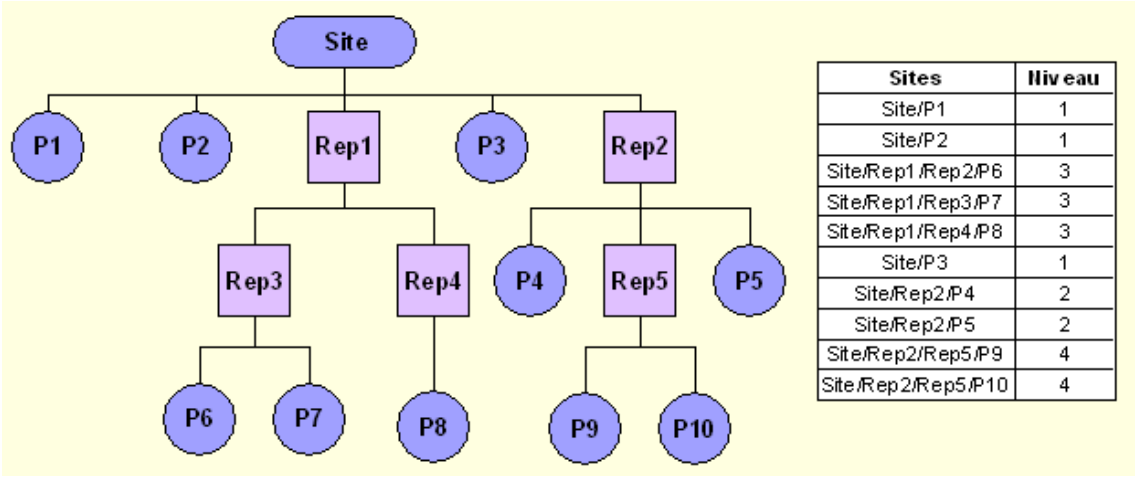
Afin de représenter le graphe du sites, nous utilisons un graphe orienté $G_s = (SW, L_s)$ dont SW est l'ensemble des super nœuds qui correspondent aux sites Web et L_s l'ensemble des super liens entre ces sites. Les caractéristiques du graphe G_s sont :

1. SW représente l'ensemble des sites Web de la collection de test. Chaque super nœud du graphe représente un site Web.
2. L_s est l'ensemble de couples ordonnés (S_i, S_j) formés de deux sites de l'ensemble SW qui matérialise l'existante d'un ou plusieurs liens hypertextes entre une page de S_i et une autre page de S_j. La représentation mathématique du graphe du site est une matrice d'adjacence MS définie comme suit :

$$MS_{i,j} = \begin{cases} 1 \; s'il \; existe \; au \; moins \; un \; lien \; entre \; une \; page \; de \; S_i \; et \; une \; page \; de \; S_j \\ 0 \; sinon \end{cases}$$

Pendant l'interrogation du système, un sous graphe de site Web $G'_s = (SW', L'_s, \omega_s)$ est associé à la requête utilisateur Q doté des caractéristiques suivantes :

1. SW' est l'ensemble des sites Web de la collection de test qui contiennent au moins une page répondant à la requête utilisateur Q. Un score de pertinence reposant sur les scores de pertinence de l'ensemble des pages du site répondant à la requête Q est associé à chaque site S de SW'. Soit $Okapi_S(S, Q)$ le score de pertinence du site S par rapport à la requête utilisateur Q reposant sur le contenu seul des pages du site S. Ce score de pertinence est calculé en tenant compte de la profondeur des pages Web dans la structure arborescente du site S. Par définition, la profondeur d'une page dans un site représente la distance qui sépare le repertoire de la page de la racine du site. La profondeur d'une page dans un site



Sites	Niveau
Site/P1	1
Site/P2	1
Site/Rep1/Rep2/P6	3
Site/Rep1/Rep3/P7	3
Site/Rep1/Rep4/P8	3
Site/P3	1
Site/Rep2/P4	2
Site/Rep2/P5	2
Site/Rep2/Rep5/P9	4
Site/Rep2/Rep5/P10	4

FIGURE 3.3: Exemple de profondeurs de pages dans un site

Web est calculée en analysant l'URL de cette page. Nous supposons que la profondeur des pages de la racine du site est égale à 1. Pour les autres pages, la profondeur indique le nombre de répertoires qui séparent la page de la racine du site. La figure 3.3 montre un exemple de profondeurs différentes des pages dans un site Web.

le score de pertinence d'un site par rapport à la requête utilisateur Q, $Okapi_S(S,Q)$, est calculé de la manière suivante :

$$Okapi_S(S,Q) = \frac{\sum_{P \in S} \frac{Okapi_D(P,Q)}{PF(P,S)}}{N_p(S,Q)} \tag{3.12}$$

avec $Okapi_D(P,Q)$ le score de pertinence de la page P reposant sur le contenu seul, $PF(P,S)$ la profondeur de de la page P dans le site S et $N_p(S,Q)$ le nombre de pages du site S répondant à la requête utilisateur Q. Nous appelons le score de pertinence d'un site la richesse d'information du site par rapport à là requête utilisateur Q. Elle représente le score de pertinence moyen des pages en réponse à la requête utilisateur Q.

2. L'_s est l'ensemble de couples ordonnés (S_i, S_j) formés de deux sites appartenant à l'ensemble de site SW'.

3. ω_s est la fonction de pondération des super liens qui associe à chaque super lien de L'_s un poids calculé en tenant compte des poids des liens hypertextes qui relient les pages de S_i à des pages de S_j. Étant donnée que les poids des liens hypertextes dépendent du nombre de termes de la requête utilisateur Q, le poids d'un super lien dépend lui aussi indirectement du nombre de termes de la requête Q. Ces poids servent à propager de la richesse d'information des sites Web par rapport à la requête utilisateur Q à travers le graphe de sites. Le poids d'un super lien est défini comme suit :

$$\begin{aligned} &\omega_{\mathrm{s}} : L'_s \rightarrow R \\ &(S_i, S_j) \rightarrow \omega_{\mathrm{s}}(S_i, S_j, Q) = \textstyle\sum_{P_x \in S_i \wedge P_y \in S_j} \omega_p\,(P_x, P_y, Q) \end{aligned}$$

Avec $\omega_p(P_x, P_y, Q)$ le poids du lien hypertexte qui relie la page P_x à la page P_y.

Une fois la richesse d'information des sites Web et les poids des super liens sont calculés, nous appliquons notre modèle générique de propagation de pertinence pour propager de la richesse d'information des sites Web

à travers les super liens du graphe de sites généré G'_s. Le voisinage d'un site est calculé de la manière suivante :

$$V_s(S,Q) = \sum_{S_i \to S} \frac{\omega_s(S_i,S,Q) * Okapi_S(S_i,Q)}{|V(S)|} \tag{3.13}$$

Avec $V(S)$ est l'ensemble des sites qui pointent le site S. $Okapi_S(S_i,Q)$ représente la richesse d'information du site S_i par rapport à la requête utilisateur Q reposant sur le contenu seul des pages du site S_i. $\omega_s(S_i,S,Q)$ est le poids du super lien qui relie les deux sites S_i et S.

Le score final de la pertinence d'un site S par rapport à la requête utilisateur Q est défini comme suit :

$$S_s(S,Q) = \lambda * Okapi_S(S,Q) + (1-\lambda) * V_s(S,Q) \tag{3.14}$$

Avec λ un paramètre compris entre 0 et 1. Ce paramètre nous permet de voir l'impact du voisinage du site sur la richesse d'informa du site. $Okapi_S(S,Q)$ la richesse d'information du site S par rapport à la requête utilisateur Q. $V_s(S,Q)$ le voisinage du site S.

Afin d'évaluer ce modèle de propagation de la richesse d'information de sites Web, nous sommes obligés de calculer la pertinence au niveau page en tenant compte de la richesse d'information des sites Web. En effet, les jugements de pertinence de nos collections de test ne prennent pas en considération la pertinence des sites. Tous les jugements ont été effectué pour le niveau page. Pour ce faire, une combinaison du score de pertinence d'une page avec le degré de la richesse d'information du site de la page permet de calculer un nouveau score de pertinence de la page par rapport à la requête utilisateur Q. Nous avons opté pour la somme des deux scores.

$$S_p^s(P_i^x,Q) = Okapi_D(P_i^x,Q) + S_s(S_x,Q) \tag{3.15}$$

Avec $Okapi_D(P_i^x,Q)$ le score de pertinence de la page P_i^x appartenant au site S_x par rapport à la requête utilisateur Q, $S_s(S_x,Q)$ le score de pertinence du site S_x de la page P_i^x par rapport à la requête utilisateur Q.

3.3.3 Niveau bloc

Le plus bas niveau est le niveau bloc qui permet de cibler l'information recherchée dans une partie de la page Web. Le web est composé de pages hétérogènes. Le multi-thèmes et la longueur des pages Web sont deux facteurs pouvant affecter d'une manière significative les performances de la recherche d'information sur le Web. En effet, une page Web contient souvent divers contenus de thématiques différentes. De plus, il existe des blocs qui n'ont rien à voir avec la recherche effectuée et qui faussent le calcul de la pertinence de la page par rapport à un besoin utilisateur. Ces blocs correspondent aux barres de navigation, de publicité et d'offres commerciales et ne doivent pas intervenir dans le calcul de la pertinence de la page. Nous supposons que le fait de découper une page Web en plusieurs blocs thématiques améliore les performances du système de recherche d'information en renvoyant en plus de la page qui contient l'information recherchée, le bloc ou les blocs susceptible de contenir plus d'information sur la requête utilisateur. La spécificité de tels blocs réside dans la cohérence des termes à l'intérieur de ces blocs. De plus, les auteurs des pages Web organisent les informations contenus dans ces pages en plusieurs blocs thématiques séparés par des délimiteurs de blocs qui corespondent aux balises HTML de la structure DOM de la page. Dans le web, une page Web est perçue comme plusieurs objets différents plutôt qu'un seul objet. Les utilisateurs distinguent entre les différents blocs d'une page Web grâce aux indicateurs visuels contenus dans cette page tels que la couleur, les lignes horizontales et verticales et la représentation structurelle de la page (paragraphes <p>, titres <H1>..<H6>). La plupart des travaux de segmentation de pages Web utilisent ces critères visuels afin de déterminer les frontières entre les différents blocs d'une page Web. Le problème qui se pose maintenant est comment choisir ces critères visuels sachant qu'il y a une multitude de

critères visuels existants et que chaque conception d'une page Web diffère de la conception d'une autre page dans le choix des délimiteurs de blocs ? Pour ce faire, Nous avons proposé une manière de découper les pages Web en plusieurs blocs thématiques en utilisant des critères visuels de segmentation et en évaluant plusieurs solutions de segmentation. Nous verrons cette technique de segmentation dans la section suivante. L'objectif d'une telle segmentation est de parvenir à découper les pages en blocs cohérents à l'intérieur de leurs contenus et distants entre les blocs adjacents.

Afin d'appliquer notre modèle générique de propagation de pertinence au niveau bloc, nous avons besoin de construire le graphe de blocs. Or, il n'existe pas de liens qui relient les blocs générés. De ce fait, nous avons construit un graphe de blocs à partir du graphe de pages. On distingue deux types de liens :

– Les liens explicites qui sont des liens hypertextes reliant des blocs à des pages. Nous avons traduit chaque lien entre un bloc et une page en plusieurs liens entre le bloc source du lien et tous les blocs de la page destinataire du lien.
– Les liens implicites qui sont des liens créés dans le but de relier les blocs de la même page Web. Ces liens seront créés pendant l'interrogation du système pour relier les blocs de la page qui répondent à la requête utilisateur.

Nous utilisons un graphe orienté $G_b = (BT, L_b)$ dont BT est l'ensemble des blocs thématiques générés en appliquant l'algorithme de segmentation que nous allons détailler dans la section suivante. L_b est l'ensemble des liens explicites et implicites qui relient des blocs de pages différentes et des blocs de la même page respectivement. Les caractéristiques du graphe G_b sont :

1. BT représente l'ensemble des blocs thématiques créés en appliquant un algorithme de segmentation de pages de la collection de test reposant sur l'évaluation de plusieurs solutions de segmentation d'une même page. Chaque nœud du graphe G_b représente un bloc thématique.
2. L_b est l'ensemble de couples ordonnés (B_i^x, B_j^y) formés de deux blocs thématiques de l'ensemble BT qui matérialise l'existante d'au moins un lien implicite ou explicite qui relie le bloc B_i^x de la page P_x au bloc B_j^y de la page P_y. Cependant, il n'existe pas des liens explicites entre blocs mais des liens qui relient des blocs à des pages. Afin de remédier à ce problème, nous avons traduit chaque lien hypertexte entre un bloc et une page en plusieurs liens entre le bloc source du lien et les blocs de la page destinataire du lien. Pour différencier ces liens entre blocs, nous avons introduit un facteur appelé importance d'un bloc dans une page. Nous supposons qu'un bloc qui contient plusieurs termes est plus important qu'un bloc qui contient moins de termes. Nous tenons compte de l'importance des blocs dans les pages dans la pondération des liens entre ces blocs. L'importance des blocs n'est autre qu'une distribution de probabilité calculée à partir des tailles des blocs et de la taille de la page en nombre de termes d'indexation qu'ils contiennent. Elle est définie comme suit :

$$\mathrm{Imp}\,(\mathrm{B_i^x}) = \frac{dl\,(B_i^x)}{dl\,(P_x)} \tag{3.16}$$

Avec B_i^x un bloc appartement à la page P_x. $dl(P_x)$ et $dl(B_i^x)$ représentent la taille de la page P_x et la taille du bloc B_i^x respectivement. La taille d'un bloc ou d'une page correspond au nombre de termes du bloc ou de la page respectivement. La représentation mathématique du graphe de blocs thématiques est une matrice d'adjacence MB définie comme suit :

$$MB_{i,j}^{x,y} = \begin{cases} 1 & \mathrm{S'il\ existe\ un\ lien\ hypertexte\ entre\ le\ bloc\ B_i^x\ et\ la\ page\ P_y\ qui\ contient\ le\ bloc\ B_j^y} \\ 0 & \mathrm{Sinon} \end{cases}$$

Pendant l'interrogation du système, un sous graphe de blocs thématiques $G'_b = (BT', L'_b, \omega_b)$ est associé à la requête utilisateur Q. Ce graphe est caractérisé par l'ensemble des propriétés suivantes :

1. BT' représente l'ensemble des blocs de BT qui répondent à la requête utilisateur Q. Ces blocs contiennent au moins un terme de la requête Q. Un score de pertinence reposant sur le contenu seul du bloc, calculé par la formule OKAPI BM25, est associé à chaque bloc de BT'. Soit $Okapi_D(B, Q)$ le score de pertinence du bloc B par rapport à la requête utilisateur Q.
2. L'_b est l'ensemble de couples ordonnés (B_i, B_j) formés de deux blocs dont l'un d'eux appartient à l'ensemble BT'. Il existe deux types de liens : les liens explicites qui sont des liens hypertextes relient des blocs appartenant à des pages différentes et les liens implicites qui sont des liens créés pendant l'interrogation du système pour relier les blocs de la même page qui répondent à la requête utilisateur Q.
3. ω_p est la fonction de pondération des liens qui associe à chaque lien de L'_b un poids calculé en tenant compte du nombre de termes de la requête Q contenu dans bloc source du lien. Cette fonction est définie comme suit :

$$\omega_{\mathrm{b}} : L'_b \rightarrow R$$

$$\left(B_i^x, B_j^y\right) \rightarrow \omega_b\left(B_i^x, B_j^y, Q\right) = \begin{cases} imp\left(B_j^y\right) * \frac{2^k}{2^{ntq+1} * \left(1-\left(\frac{1}{2}\right)^{ntq}\right)} & \text{Si x} \neq \text{y} \\ \frac{2^k}{2^{ntq+1} * \left(1-\left(\frac{1}{2}\right)^{ntq}\right)} & \text{Si x} = \text{y} \end{cases}$$

Avec B_i^x, B_j^y deux blocs appartenant aux pages P_x et P_y respectivement. k représente le nombre de termes de la requête Q contenu dans le bloc B_i^x. ntq est le nombre de termes de la requête Q. Dans le premier cas, le lien est explicite (les deux blocs appartiennent à deux pages différentes relié par un lien hypertexte). Tandis que le lien dans le deuxième cas est implicite (les deux blocs appartiennent à la même page et répondent à la requête utilisateur Q).

Une fois les scores de pertinence des blocs thématiques et les poids des liens explicites et implicites qui relient ces blocs sont calculés, nous appliquons notre modèle générique de propagation de pertinence pour propager les scores de pertinence des blocs à travers les liens explicites et implicites du graphe de blocs généré G'_b. Le voisinage d'un bloc est calculé comme suit :

$$V_b\left(B_i^x, Q\right) = \sum_{B_j^y \rightarrow B_i^x} \frac{\omega_b\left(B_j^y, B_i^x, Q\right) * Okapi_D\left(B_j^y, Q\right)}{|V\left(B_i^x\right)|} \tag{3.17}$$

Avec $V(B_i^x)$ l'ensemble des blocs qui pointent le bloc B_i^x. $Okapi_D(B_j^y, Q)$ représente le score de pertinence reposant sur le contenu seul du bloc B_j^y appartenant à la page P_y par rapport à la requête utilisateur Q. $\omega_s(B_j^y, B_i^x, Q)$ est le poids du lien implicite ou explicite qui relie le bloc B_j^y de la page P_y au bloc B_i^x de la page P_x.

Le score final de la pertinence d'un bloc B_i^x par rapport à la requête utilisateur Q est défini comme suit :

$$S_b\left(B_i^x, Q\right) = \gamma * Okapi_D\left(B_i^x, Q\right) + (1-\gamma) * V_b\left(B_i^x, Q\right) \tag{3.18}$$

Avec γ un paramètre compris entre 0 et 1. Ce paramètre nous permet de voir l'impact du voisinage de la page sur la pertinence de la page basée sur le contenu seul. $Okapi_D(B_i^x, Q)$ représente le score de pertinence reposant sur le contenu seul de du bloc B_i^x appartenant à la page P_x par rapport à la requête utilisateur Q. $V_b(B_i^x, Q)$ le score de voisinage du bloc B_i^x de la page P_x par rapport à la requête utilisateur Q calculé en pondérant les liens qui relient ces blocs en fonction du nombre de termes de la requête Q contenu dans ces blocs.

Afin d'évaluer le modèle de propagation de pertinence au niveau bloc, nous sommes obligés de calculer la pertinence au niveau page en tenant compte de la pertinence des blocs appartenant à cette page. En effet,

les jugements de pertinence de nos collections de test ne prennent pas en considération la pertinence des blocs thématiques. Tous les jugements ont été effectué pour le niveau page. Pour ce faire, le score de pertinence d'une page est calculé en fonction des scores de pertinence des blocs de la page. Nous avons opté pour la somme des scores de pertinence des blocs de la page qui répondent à la requête utilisateur de la manière suivante :

une combinaison du score de pertinence d'une page avec le degré de la richesse d'information du site de la page permet de calculer un nouveau score de pertinence de la page par rapport à la requête utilisateur Q. Nous avons opté pour la moyenne des deux scores.

$$S_p^b\left(P_x, Q\right) = \sum_{B_i^x \in P_x} S_b\left(B_i^x, Q\right) \tag{3.19}$$

Avec $S_p^b(P_x, Q)$ le score de pertinence de la page Px par rapport à la requête utilisateur Q, $S_b(B_i^x, Q)$ le score de pertinence du bloc B_i^x de la page P_x par rapport à la requête utilisateur Q.

3.4 Algorithme de segmentation

Une analyse thématique fondée sur la répartition des termes dans un texte part d'un constat que le développement d'un thème entraîne la reprise de termes spécifiques. La reconnaissance de parties de la page liées à un même sujet est alors fondée sur la distribution des termes et leurs récurrences. Si un terme apparaît souvent dans l'ensemble de la page, il est peu significatif, alors que sa répétition dans une partie limitée de la page est très significative pour caractériser le thème du bloc qu'il le contient. En plus, lorsqu'un auteur traite un sujet, il en expose en général un point de vue, en développant des aspects particuliers, ce qui conduit à délimiter des blocs distincts. Cependant les enchaînements possibles de différents blocs suivent des critères forts de cohérence. Le découpage d'une page en plusieurs blocs thématiques repose sur des critères de délimitations de blocs qui permettent de fractionner la page Web à des endroits spécifiques. Ces critères ne sont que les balises HTML de la page qui décrivent la structure de la page. Le problème majeur de la segmentation d'une page Web est le fait qu'il existe différentes manières de découper une page Web en plusieurs blocs selon différents critères susceptible d'être des critères de délimitation de blocs.

Dans ce qui suit, nous proposons une solution pour la segmentation d'une page Web. Cette solution repose sur un algorithme génétique. Le but est de trouver une segmentation à base des critères visuels (ligne, la couleur) et de représentation du contenu (paragraphe, sous-titres) qui permet d'avoir des blocs thématiquement homogènes. Ces critères de délimitation de blocs permettent dans la plupart des cas de passer d'un bloc à un autre ou de changer une idée exposée dans un bloc précédent. Nous voulons combiner ces deux critères afin de segmenter les pages Web de sorte que les blocs soient distants entre eux et homogènes à l'intérieur de leur contenu. Notre travail consiste alors à extraire des blocs à partir des pages Web en utilisant la structure HTML de la page (les critères visuels et de présentation).

Afin de pouvoir calculer ces distances, nous disposons de deux mesures : une est appliquée à l'intérieur d'un bloc qui repose sur la co-occurrence entre les termes appartenant au même bloc. Et l'autre se base sur la mesure du cosinus entre deux vecteurs blocs. Avant de proposer notre solution, une brève description des algorithmes génétiques sera présentée dans la section suivante :

3.4.1 Algorithmes génétiques

Les algorithmes génétiques permettent d'obtenir une solution approchée, en un temps correct, à un problème d'optimisation, lorsqu'il n'existe pas de méthodes exactes pour le résoudre. Les algorithmes génétiques

utilisent la notion de sélection naturelle développée au 20^ime siècle par le scientifique Charles Darwin et l'appliquent à une population de solutions potentielles au problème donné. En effet, les individus les plus adaptés tendent à survivre plus longtemps et à se reproduire plus aisément. Charles Darwin a observé les phénomènes naturels et a fait les constatations suivantes :

- l'évolution n'agit pas directement sur les êtres vivants, elle opère en réalité sur les chromosomes contenus dans leur ADN.
- l'évolution a deux composantes : la sélection qui garantit une reproduction plus fréquente des chromosomes les plus forts et la reproduction qui est la phase durant laquelle s'effectue l'évolution.

Dans les années 60s, John H. Holland [Hol62] a expliqué comment ajouter de l'intelligence dans un programme informatique avec les croisements et la mutation. Le croisement est l'opérateur de l'algorithme génétique qui permet le plus souvent de se rapprocher de l'optimum d'une fonction en combinant les gènes contenus dans les différents individus de la population. Le premier aboutissement des travaux de recherches de Holland est la formalisation des principes fondamentaux des algorithmes génétiques (Holland [Hol75]) surtout la capacité de représentations élémentaires, comme les chaînes de bits, à coder des structures complexes.

Les algorithmes génétiques étant basés sur des phénomènes biologiques, il convient de rappeler au préalable quelques termes de génétique. Les organismes vivants sont tout d'abord constitués de cellules comportant des chromosomes qui correspondent en fait à des chaînes d'ADN. L'élément de base de ces chromosomes (le caractère de la chaîne d'ADN) est un gène. Sur chacun de ces chromosomes, une suite de gènes constitue une chaîne qui code les fonctionnalités de l'organisme (la couleur des yeux, la couleur de la peau, la taille, etc.). On utilise aussi, dans les algorithmes génétiques, une analogie avec la biologie, qui concerne l'évolution, hypothèse émise par Darwin et qui propose qu'au fil du temps, les gènes conservés au sein d'une population donnée sont ceux qui sont les plus adaptés aux besoins de l'espèce et à son environnement.

Un algorithme génétique recherche le ou les extremums d'une fonction définie sur un espace de données. Pour l'utiliser, on doit disposer des cinq éléments suivants :

1. Un principe de codage des individus d'une population. Cette étape associe à chacun des points de l'espace d'état une structure de données. Elle se place généralement après une phase de modélisation mathématique du problème traité. La qualité du codage des données conditionne le succès des algorithmes génétiques et leurs convergences. Les codages binaires sont certainement les plus utilisés car il présentent plusieurs avantages. Le principe d'un codage binaire est de coder la solution selon une chaîne de bits (qui peuvent prendre les valeurs 0 ou 1). Les raisons pour lesquelles ce type de codage est le plus utilisé sont tout d'abords historiques. En effet, lors des premiers travaux de Holland [Hol62], les théories ont été élaborées en se basant sur ce type de codage. Et même si la plupart de ces théories peuvent être étendues à des données autres que des chaînes de bits, elles n'ont pas été autant étudiées dans ces contextes. Il existe cependant au moins un coté négatif à ce type de codage qui fait que d'autres existent. En effet, ce codage est souvent peu naturel par rapport à un problème donné. Parmi les autres codages existants, nous distinguons deux types de codage qui peuvent nous servir dans nos travaux de thèse : le codage à caractères multiples et le codage sous forme d'arbre. Le premier codage code les chromosomes d'un algorithme génétique à l'aide de caractères multiples. Souvent, ce type de codage est plus naturel que celui du codage binaire. C'est d'ailleurs celui-ci qui est utilisé dans de nombreux cas poussés d'algorithmes génétiques. Le deuxième codage utilise une structure arborescente avec une racine de laquelle peuvent être issus un ou plusieurs fils. Un de leurs avantages est qu'ils peuvent être utilisés dans le cas de problèmes où les solutions n'ont pas une taille finie. En principe, des arbres de taille quelconque peuvent être formés par le biais de croisements et de mutations. Le problème de ce type de codage est que les arbres résultants sont souvent difficiles à analyser et que l'on peut se retrouver avec des arbres solutions dont la taille sera importante alors qu'il existe des solutions plus simples et plus structurées à côté desquelles sera passé l'algorithme. De plus, les performances de ce type de codage par rapport à des codages en chaînes n'ont pas encore été comparées ou très peu. En effet, ce type d'expérience ne fait

que commencer et les informations sont trop faibles pour se prononcer. Le choix du type de codage ne peut pas être effectué de manière sûre dans l'état actuel des connaissances. Selon les chercheurs dans ce domaine, la méthode actuelle à appliquer dans le choix du codage consiste à choisir celui qui semble le plus naturel en fonction du problème à traiter et développer ensuite l'algorithme de traitement.

2. Un mécanisme de génération de la population initiale. Ce mécanisme doit être capable de produire une population d'individus non homogène qui servira de base pour les générations futures. Le choix de la population initiale est important car il peut rendre plus ou moins rapide la convergence vers l'optimum global. Dans le cas où l'on ne connaît rien du problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche.
3. Une fonction à optimiser. Celle-ci retourne une valeur d'adaptation de la solution appelée fitness ou fonction d'évaluation de l'individu.
4. Des opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace de solutions. L'opérateur de croisement recompose les gènes d'individus existant dans la population. En effet, lors de l'opération de croisement, deux chromosomes s'échangent des parties de leurs chaînes, pour donner de nouveaux chromosomes. Ces croisements peuvent être simples ou multiples. Dans le premier cas, les deux chromosomes se croisent et s'échangent des portions d'ADN en un seul point. Dans le deuxième cas, il y a plusieurs points de croisement. Pour les algorithmes génétiques, c'est cette opération (le plus souvent sous sa forme simple) qui est prépondérante. Sa probabilité d'apparition P_c lors d'un croisement entre deux chromosomes est un paramètre de l'algorithme génétique. En règle générale, on fixe la probabilité d'apparition P_c aux alentours de $0,7$. L'opérateur de mutation a pour but de garantir l'exploration de l'espace de solutions. De façon aléatoire, un gène peut, au sein d'un chromosome être substitué à un autre. De la même manière que pour les croisement, on définit ici un taux de mutation P_m lors des changements de population qui est généralement faible compris entre $0,01$ et $0,1$. Il est nécessaire de choisir pour ce taux une valeur relativement faible de manière à ne pas tomber dans une recherche aléatoire et conserver le principe de sélection et d'évolution. La mutation sert à éviter une convergence prématurée de l'algorithme. Par exemple, lors d'une recherche d'extremum, la mutation sert à éviter la convergence vers un extremum local.
5. Un mécanisme de sélection des individus qui forment la nouvelle génération en ce basant sur la valeur d'adaptation de chaque solution. Ce processus est analogue à un processus de sélection naturelle, les individus les plus adaptés gagnent la compétition de la reproduction tandis que les moins adaptés meurent avant la reproduction, ce qui améliore globalement l'adaptation.
6. Des paramètres de dimensionnement : taille de la population, nombre total de générations ou critère d'arrêt, probabilités d'application des opérateurs de croisement et de mutation. Ces différents paramètres sont nécessaires pour n'importe quel algorithme génétique. Elle représentent les condition du bon fonctionnement de l'algorithme. Un choix judicieux de ces paramètres permet une convergence rapide vers la meilleure solution.

Le principe général du fonctionnement d'un algorithme génétique est représenté sur la figure 3.5.

3.4.2 Principe de notre algorithme

Notre problème est d'optimiser la segmentation d'une page Web de façon à ce que celle-ci permet de délimiter des différents blocs thématiques. Le tableau 3.1 montre comment nous avons traduit le problème de segmentation de pages Web en un problème d'optimisation de la solution de segmentation d'une page. Afin de résoudre ce problème, nous avons recours aux algorithmes génétiques. De fait qu'on ne sait pas à l'avance, quels seront les blocs à retenir et comment le faire. Les algorithmes génétiques nous permettent de se rapprocher

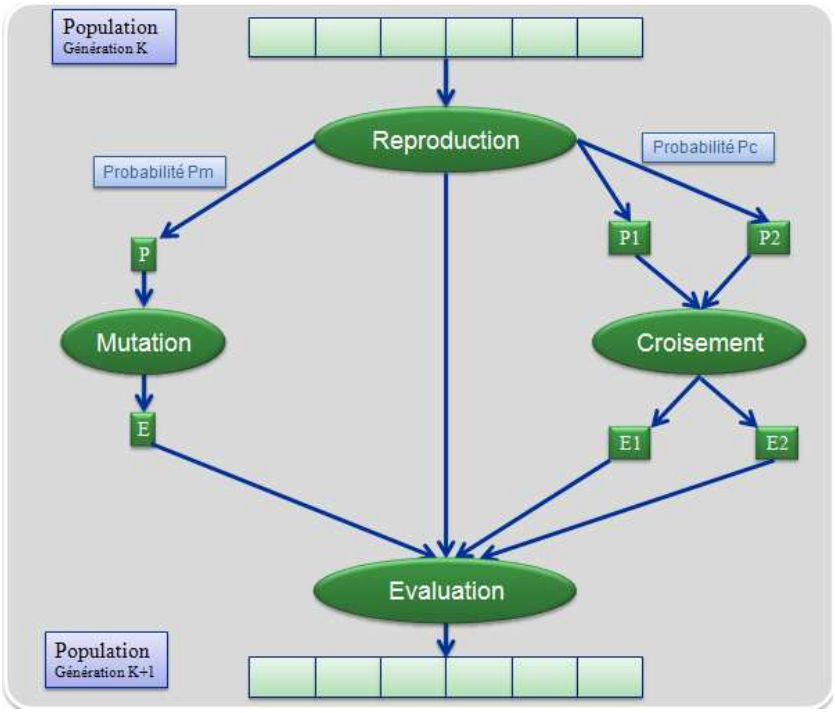


FIGURE 3.4: Principe général des algorithmes génétiques

Optimisation	Page Web
Espace de recherche	Ensemble des solutions de segmentation d'une page Web
Fonction d'évaluation	Une combinaison de la cohérence des termes à l'intérieur des blocs et de la distance entre les blocs
Solution optimale	Segmentation maximisant la cohérence des termes à l'intérieur des blocs, ainsi que la distance entre les blocs
Relation de voisinage	Définie par les séparateurs de blocs (balises HTML)

TABLE 3.1: Modélisation du problème de segmentation des pages Web comme un problème d'optimisation

de la solution optimale à notre problématique. Ces algorithmes se basent sur les différents principes décrits ci-dessus. De manière globale, on commence avec une population de base qui se compose le plus souvent de chaînes de blocs séquentiels correspondant chacune à un chromosome. Le contenu de cette population initiale est généré aléatoirement. Nous reviendrons par la suite sur la façon de choisir cette population de départ, car un bon choix de la population initiale accélère la convergence de notre algorithme. Nous allons décrire les différentes étapes de notre algorithme de segmentation dans les sections suivantes. Nous commençons par le codage de la solution proposée qui est une étape importante, si ce n'est la plus importante dans les algorithmes génétiques..

3.4.2.1 Codage de notre algorithme génétique

Dans notre système, un chromosome correspond à une page HTML et un gène correspond à un bloc de la page. Nous avons opté pour le codage binaire pour deux raisons : sa simplicité de mise en œuvre et sa ressemblance naturelle à notre problème de segmentation. En effet, la taille d'un code d'une segmentation est proportionnelle au nombre de critères de délimitation de blocs candidat à la segmentation.chaque occurrence

d'un critère de délimitation de blocs est représenté par bit. Par exemple, le premier bit correspond au premier critère de délimitation de blocs. Par conséquent, un bit à 1 dans le code signifie que le critère correspondant à ce bit est pris en considération dans la solution de segmentation, tandis qu'un bit à 0 signifie que le critère correspondant à ce bit n'est pas pris en compte dans la segmentation. La justification du codage binaire de notre algorithme est motivée par le fait que la structure d'une page Web peut être conçue comme une succession de blocs séquentiels délimité par des critères visuels. Toutefois, pour calculer la valeur d'adaptation de chaque solution de segmentation dans l'espace des solutions, nous avons besoin de calculer la coherence à l'intérieur des blocs délimités par des critères visuels et la distance entre les vecteurs de blocs adjacents. Or, le codage binaire tout seul ne nous permet pas de distinguer les différents blocs d'une solution de segmentation. Afin de garder une trace des différents blocs d'une solution de segmentation identifiée par un code binaire, nous construisons la table des blocs qui contient les différents blocs élémentaires (contenu des blocs). Par consequent, la construction d'une solution de segmentation à un instant donné à partir du codage binaire de la solution et la table des blocs est facilitée par l'application de ces règles :

- Un n^{ime} bit mis à 1 du codage binaire d'une solution indique que le n^{ime} délimiteur de blocs qui sépare le bloc n du bloc $n+1$ de la table des blocs est pris en compte dans la segmentation de la page.
- La succession de bits à 0 indique une fusion des blocs séparés par les délimiteurs correspondant à ces bits dans le code de la solution.

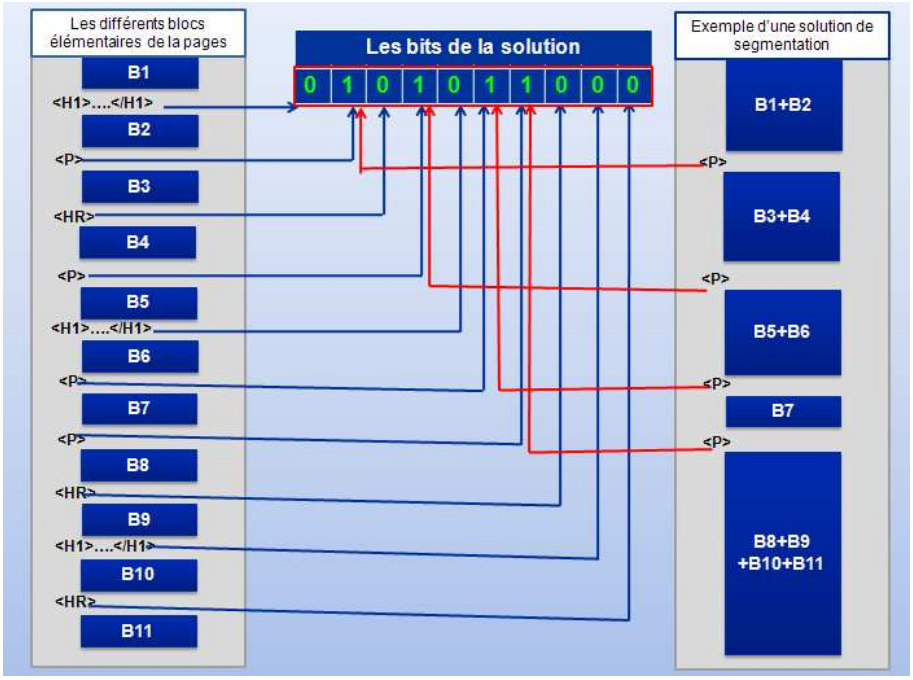


FIGURE 3.5: Exemple de codage binaire d'une solution de segmentation

Le problème qui se pose est comment choisir la liste des délimiteurs de blocs à critères visuels ? En effet, il existe plusieurs balises candidates au découpage des pages Web. Cependant, elles ne sont pas toutes des délimiteurs de blocs dans une page Web. On distingue pae exemple des balises à critères de présentation de la page tel que le gras, italique et le souligné et des balises à critères visuels qui permettent à l'utilisateur de distinguer les différents blocs de la page Web. Afin de remédier à ce problème, nous avons deux solutions :

1. la première solution est locale. Elle repose sur les différents balises HTML de la page Web à segmenter. Ce qui veut dire de recenser tous les critères qui peuvent être considérés comme des délimiteurs de blocs. Donc, il est nécessaire de choisir les critères qui vérifient certaines conditions. Par exemple, un critère qui figure rarement ou fréquemment dans la page ne doit pas être considéré comme un délimiteur de blocs. Or, la fréquence d'une balise dans une page n'est pas un indice fiable pour considérer cette balise

comme un délimiteur de blocs. De ce fait, la solution locale peut introduire du bruit dans la segmentation des pages.

2. la deuxième solution est globale. Elle consiste à calculer un poids pour chaque balise susceptible d'être choisie comme un délimiteur de blocs dans un ensemble de pages choisi aléatoirement. Ce poids est calculé en fonction de la fréquence de la balise dans chaque page de l'ensemble des pages et du nombre de pages qui contient cette balise.

Notre choix porte sur la deuxième solution. Nous envisageons de choisir une liste representative des balises HTML correspondant aux différents délimiteurs de blocs de façon à favoriser les balises courantes dans l'ensemble des pages et dont la fréquence d'apparition dans ces pages est grande. Pour cela, nous avons étudié la plupart des balises HTML les plus utilisées et nous avons calculé pour chaque balise son poids par rapport à un ensemble de pages que nous avons choisi aléatoirement. Cet ensemble contient 2500 pages de la collection WT10g. Nous avons classé les différentes balises selon leurs poids à l'intérieur de cet ensemble. Le poids d'une balise HTML est calculé de la manière suivante :

soit g une balise HTML et S un ensemble de 2500 pages choisi au hasard parmi l'ensemble des pages de la collection WT10g de TREC. $Freq(g, S)$ représente la fréquence de la balise g dans l'ensemble S, $nbp(S)$ le nombre de pages de l'ensemble S, $nbp(g, S)$ le nombre de pages de S qui contiennent la balise g et $nb(g, p)$ le nombre de fois que la balise g figure dans la page p. Nous considérons qu'une balise g appartient à une page p si et seulement si la balise g figure au moins deux fois dans la page. Cette restriction permet de ne pas tenir compte des balises qui figure une seule fois et dans toutes les pages comme les balises <HTML>, <TITLE>, <BODY>, etc. La formule mathématique du calcul du poids des balises HTML est décrite comme suit :

$$Freq(g, S) = \frac{nbp(S)}{nbp(g, S)} * \log\left(\frac{\sum\limits_{P \in S} nb(g, p)}{nbp(g, S)}\right) \tag{3.20}$$

Après avoir calculé un poids pour chaque balise HTML, nous choisissons pour notre algorithme de segmentation les balises dont le poids est élevé. Le tableau suivant montre le classement des balises HTML considérées comme des délimiteurs de blocs. Dans nos expérimentations, nous avons pris que les 5 balises les mieux classées (<HR>, <H1>...<H6>, <B>, <BR> et <P>) pour des considerations de complexité des calculs.

Après avoir coder les solutions de segmentation, nous passons à la première étape de notre algorithme génétique qui est la sélection de la population initiale. Comme on l'a dit auparavant, le choix de cette population est très important pour la convergence de notre algorithme.

3.4.2.2 Sélection de la population initiale

Nous supposons que la répétition d'un critère de délimiteur de blocs peut être vue comme un moyen de segmentation de la page Web. D'où l'idée de construire une solution par critère de délimitation. En effet, la population initiale est sélectionnée en fonction du nombre de critères de délimitation figurant dans la page Web. Par conséquent, une solution est générée pour chaque critère retenue de la façon suivante : il suffit donc de mettre tous les bits correspondant à un critère candidat à la délimitation à différent endroits de la chaîne à 1 et les autres bits correspondant à d'autres critères retenus à 0. Cependant si le nombre de critères candidats à la délimitation est insuffisant ou faible, des solutions aléatoires combinant les différents critères peuvent être générées. De ce fait, le nombre de solutions de départ qui est un paramètre de notre algorithme génétique de segmentation vaut le nombre de délimiteurs de blocs différents de la liste figurant dans la page. Une fois la population initiale est construite, nous attribuons à chacune des solutions de segmentation une valeur qui correspond à son adaptation à notre problème de segmentation. Pour cela, nous avons recours à deux mesures différentes pour calculer cette valeur d'adaptation.

Segment delimiters	Nbp(g,S)	$\sum_{PeS} nb(g,p)$	Freq(g,S)
<DIV>	1	6	0,0036
<DIR>	1	10	0,0046
<TABLE>	3	18	0,0107
<OL>	25	105	0,0717
<TD>	47	1243	0,3078
<UL>	83	636	0,3380
<TR>	75	1052	0,3961
<STRONG>	70	6359	0,6312
<HR>	220	2096	0,9918
<H1>..<H6>	210	2352	1,0146
<B>	258	6937	1,6984
 	400	20482	3,1486
<P>	458	38727	4,0646

TABLE 3.2: Classement des balises HTML selon leur poids

3.4.2.3 Le calcul de la valeur d'adaptation

Nous cherchons une fonction d'évaluation d'une segmentation qui maximise la cohérence entre les termes d'un même bloc et la distance entre des blocs adjacents. Cette fonction repose sur deux mesures : la cohérence à l'intérieur d'un bloc et la distance entre deux blocs adjacents.

3.4.2.3.1 Mesure de cohérence d'un bloc : On part d'un constat que les termes les plus fréquents dans un bloc constituent le thème du bloc. Dans nos experimentations, nous admettons que la taille minimale d'un bloc est de 8 termes et si un bloc contient moins de 8 termes, on associe ces termes au bloc qui le précède. Le choix de la taille minimale d'un bloc est arbitraire. Ce choix nous permet d'avoir des blocs consistants. La mesure de cohérence est appliquée à l'intérieur d'un bloc et dépend de la matrice des co-occurrences entre les termes d'indexation des documents. Elle reflète la densité des informations relatives à un thème particulier et le degré de corrélation entre les termes du bloc. Nous n'avons pas tenu compte de tous les termes dans le calcul de la coherence d'un bloc, mais uniquement les 8 termes les plus fréquents. Cela nous permet d'accélérer, d'une part, les calculs et de garantir, d'autre part, l'équité du traitement des blocs qui ont des tailles différentes en nombre de termes. La cohérence d'un bloc est calculé de la manière suivante :

$$COH_{int}(B) = \frac{1}{8^2} * \sum_{T_i \in B} \sum_{T_j \in B} CC(T_i, T_j) * \frac{TF(T_i, B) * TF(T_j, B)}{\left(\underset{T_k \in B}{MAX} (TF(T_k, B)) \right)^2} \tag{3.21}$$

Avec $TF(T_k, B)$ le nombre d'occurrences du terme T_k dans le bloc B. $k \in [1, 8]$ et correspond aux 8 termes les plus fréquents dans le bloc B. $CC(T_i, T_j)$ désigne la co-occurrence entre les deux termes T_i et T_j. Elle représente la fraction du nombre de documents qui contiennent les deux termes T_i et T_j par rapport au nombre de documents qui contiennent au moins l'un des deux termes. La co-occurrence entre deux termes est définie comme suit :

$$CC(T_i, T_j) = \begin{cases} \frac{|E_d(T_i) \cap E_d(T_j)|}{|E_d(T_i) \cup E_d(T_j)| - |E_d(T_i) \cap E_d(T_j)|} & \text{SiT}_\text{i} \neq \text{T}_\text{j} \\ 1 \quad Sinon \end{cases} \tag{3.22}$$

Dans l'équation 3.22, $E_d(T_k)$ représente l'ensemble des documents qui contiennent le terme T_k et $|E_d(T_k)|$ la taille de cet ensemble.

Nous remarquons que plus la co-occurrence entre les termes d'un bloc est grande, plus la cohérence à l'intérieur du bloc est élevée.

3.4.2.3.2 Mesure de distance entre deux blocs adjacents : Il n'existe pas une vraie distance entre deux vecteurs blocs. Dans notre système, nous avons calculé une valeur qui peut être interprétée comme une distance entre deux blocs. Cette mesure est basée sur la similarité et la coherence des termes entre deux blocs adjacents. En effet, la mesure de similarité entre blocs repose sur les termes en communs entre deux blocs adjacents, tandis que la cohérence entre deux blocs repose sur le degré de correlation entre les termes des deux blocs. Cette dernière permet de calculer une certaine similarité entre deux blocs en tenant compte de la similarité entre les termes des deux blocs calculée en fonction de la co-occurrence entre les termes les plus fréquents de ces deux blocs. La distance entre deux blocs adjacents est définie comme suit :

$$DIST\left(B_x, B_y\right) = \left(1 - SIM\left(B_x, B_y\right)\right) * \left(1 - COH_{adj}\left(B_x, B_y\right)\right) \tag{3.23}$$

La similarité entre deux blocs adjacents est calculée en tenant compte de tous les termes des deux blocs. Elle est calculée de la manière suivante :

$$SIM(B_x, B_y) = \frac{\sum\limits_{T_k \in B_x \cap B_y} TF(T_k, B_x) * TF(T_k, B_y)}{\sqrt{\sum\limits_{T_i \in B_x} (TF(T_i, B_x))^2} * \sqrt{\sum\limits_{T_j \in B_y} (TF(T_j, B_y))^2}} \tag{3.24}$$

La cohérence entre deux blocs adjacents est décrite comme suit :

$$COH_{adj}\left(B_x, B_y\right) = \frac{1}{8^2} * \sum_{T_i \in B_x} \sum_{T_j \in B_y} CC\left(T_i, T_j\right) * \frac{TF\left(T_i, B_x\right) * TF\left(T_j, B_y\right)}{\underset{T_k \in B_x}{MAX}\left(TF\left(T_k, B_x\right)\right) * \underset{T_m \in B_y}{MAX}\left(TF\left(T_m, B_y\right)\right)} \tag{3.25}$$

Avec $TF(T_i, B_x)$ et $TF(T_j, B_y)$ le nombre d'occurrences du terme T_i et T_j dans les blocs B_x et B_j respectivement. $i, j \in [1, 8]$ et correspondent aux 8 termes les plus fréquents dans les blocs B_x et B_y respectivement. $CC(T_i, T_j)$ désigne la co-occurrence entre les deux termes T_i et T_j décrite dans l'equation 3.22.

3.4.2.3.3 Fonction d'évaluation d'une segmentation : La fonction d'évaluation d'une segmentation est calculée à partir des deux mesures : la cohérence du contenu des blocs d'une page et la distance entre ces blocs. Nous avons opté pour la multiplication entre ces deux mesures pour avoir une valeur d'evaluation comprise entre 0 et 1. Cette mesure est décrite de la manière suivante :

$$\mathrm{Eval_{segm}}\left(\mathrm{S_i}, P\right) = \left[\frac{1}{N_i} \sum_{k=1}^{N_i} COH_{\mathrm{int}}\left(B_{i,k}\right)\right] * \left[\frac{1}{N_i - 1} \sum_{k=1}^{N_i - 1} DIST\left(B_{i,k}, B_{i,k+1}\right)\right] \tag{3.26}$$

Avec S_i une solution de segmentation de la page P reposant sur le i^{me} délimiteur de blocs (chaque solution de segmentation repose sur un critère visuel candidat à la segmentation). N_i représente le nombre de blocs thématiques resultant de la segmentation de la page P en utilisant le i^{me} délimiteur de blocs. $B_{i,k}$ le k^{me} bloc de la solution S_i. La meilleure solution de segmentation est celle qui a une grande valeur de la fonction $Eval_{segm}(S_i, P)$. C'est cette solution qui sera retenue afin de segmenter la page P.

Une fois la population initiale est construite et les valeurs d'adaptation des différents solutions sont calculé, nous effectuons des opérations de l'algorithme génétique (sélection de la nouvelle population, croisement et mutation) au sein de cette population. Nous débutons par la sélection des individus apte à se reproduire. Le processus de sélection des individus de la nouvelle génération, nous permet d'avoir une nouvelle génération meilleure que l'ancienne génération.Par conséquent, le temps d'exécution d'un tel algorithme peut prendre beaucoup de temps et l'optimum n'est pas atteint dans la plupart des cas. Dans la pratique, il est évident que le nombre de pages à segmenter est très important, d'où la nécessite de simplifier l'algorithme de segmentation. Pour ce faire, nous avons limité le nombre de critères par solution de segmentation à un seul critère de délimitation de blocs. L'algorithme est appliqué une seule fois et la meilleure solution est retenue pour le découpage de la page en bloc thématique.

3.4.2.4 la sélection de la nouvelle génération

Il existe plusieurs techniques de sélection. Voici les principales utilisées :

1. **Sélection par rang :** Cette technique de sélection choisit toujours les individus possédant les meilleurs scores d'adaptation, le hasard n'entre donc pas dans ce mode de sélection.
2. **Probabilité de sélection proportionnelle à l'adaptation :** Appelé aussi roue de la fortune. Pour chaque individu, la probabilité d'être sélectionné est proportionnelle à son adaptation au problème. Afin de sélectionner un individu, on utilise le principe de la roue de la fortune biaisée. Cette roue est une roue de la fortune classique sur laquelle chaque solution de segmentation est représentée par une portion proportionnelle à son adaptation. On effectue ensuite un tirage au sort homogène sur cette roue.
3. **Sélection par tournoi :** Cette technique utilise la sélection proportionnelle sur des paires d'individus, puis choisit pour ces paires l'individu qui a le meilleur score d'adaptation.
4. **Sélection uniforme :** La sélection se fait aléatoirement, uniformément et sans intervention de la valeur d'adaptation. Chaque solution de segmentation a donc une probabilité $1/P$ d'être sélectionné, où P est le nombre total de solution de segmentation dans la population.

La technique de selection la plus adaptée à notre algorithme de segmentation est la premiere solution. Celle-ci nous permet de donner plus d'importance aux solutions les plus adaptées à notre problème de segmentation. Par conséquence, les meilleures solutions de segmentation parmi l'ancienne et la nouvelle generation seront sélectionnées.

3.4.2.5 Les opérateurs de l'algorithme de segmentation

On distingue les deux opérations utilisées dans les algorithmes génétiques. Le croisement et la mutation. Ces deux opérations nous permettent la reproduction de notre population d'une manière à ce que la nouvelle population soit meilleure que l'ancienne. Le croisement s'applique à des paires de solutions de segmentation afin d'obtenir deux nouvelles solutions de segmentation selon le mode de croisement effectué (simple ou multiple). Nous effectuons ensuite des mutations sur une faible proportion de solutions de segmentation choisit aléatoirement (une seule solution). Ce processus nous fournit une nouvelle population. On réitère le processus un grand nombre de fois de manière à imiter le principe d'évolution, qui ne prend son sens que sur un nombre important de générations. On peut arrêter le processus au bout d'un nombre arbitraire de générations ou lorsque qu'une solution possède une valeur d'adaptation suffisamment satisfaisante.

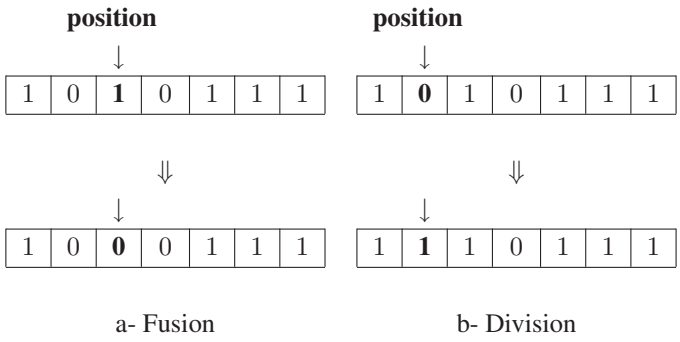
3.4.2.5.1 Le croisement Le croisement s'effectue sur deux solutions différentes de segmentation et il nous permet d'obtenir deux solutions enfants. Nous avons opté pour le croisement multiple à deux points. Le principe d'un tel croisement est le suivant : Étant donné deux solutions de segmentation pères. Dans un premier lieu, on choisit deux points de croisement aléatoirement. Puis, on combine les deux solutions pères pour construire deux autres. Nous allons suivre la méthode suivante : On choisi aléatoirement deux points de découpe. On interverti, entre les deux solutions, les parties qui se trouvent entre ces deux points comme le montre la figure ci-dessous :

Solution	**Code**	**Solution**	**Code**
Pères			
A	**10 :110010**	A	**10 : 110 :010**
B	**01 :010100**	B	**01 : 010 :100**
Enfants			
A'	**01 : 110010**	A'	**10 : 010 :010**
B'	**10 : 010100**	B'	**10 : 110 :100**

a) croisement simple **b) croisement multiple à deux points**

FIGURE 3.6: Exemple de croisement entre deux solutions de segmentation

3.4.2.5.2 La mutation Par définition, la mutation s'agit de modifier un des éléments constitutifs de la solution, dans notre cas, un délimiteur. Quand un délimiteur doit être muté, nous inversons son bit correspondant dans la chaîne. Nous distinguons deux types de mutation selon le bit résultant de la mutation. Le premier type correspond à la fusion de deux blocs. Il s'obtient en inversant le bit à 1 d'un délimiteur en un bit à 0. Le deuxième type de la mutation concerne la division. Il correspond à l'inversion d'un bit à 0 en un bit à 1. Le délimiteur correspondant au bit 0 inversé serra le nouveau délimiteur qui devise le bloc en deux. Le choix du bit à inverser est aléatoire et le nombre de solutions à muter ne dois pas dépasser un seuil que nous fixions à l'avance. Dans notre cas, une seule solution serra mutée.



a- Fusion b- Division

FIGURE 3.7: Exemple de mutation de deux solutions de segmentation

En effectuant ces deux opérations (le croisement et la mutation) un nombre de fois correspondant à la taille de la population divisée par deux, on se retrouve alors avec une nouvelle population ayant la même taille que la population initiale, et qui contient globalement des solutions plus proches de l'optimum. Le principe des algorithmes génétiques est d'effectuer ces opérations un maximum de fois de façon à augmenter la justesse du résultat. Le critère d'arrêt de notre algorithme est le nombre d'itération à executer. Dans ce qui suit, nous détaillons le processus de segmentation simplifié.

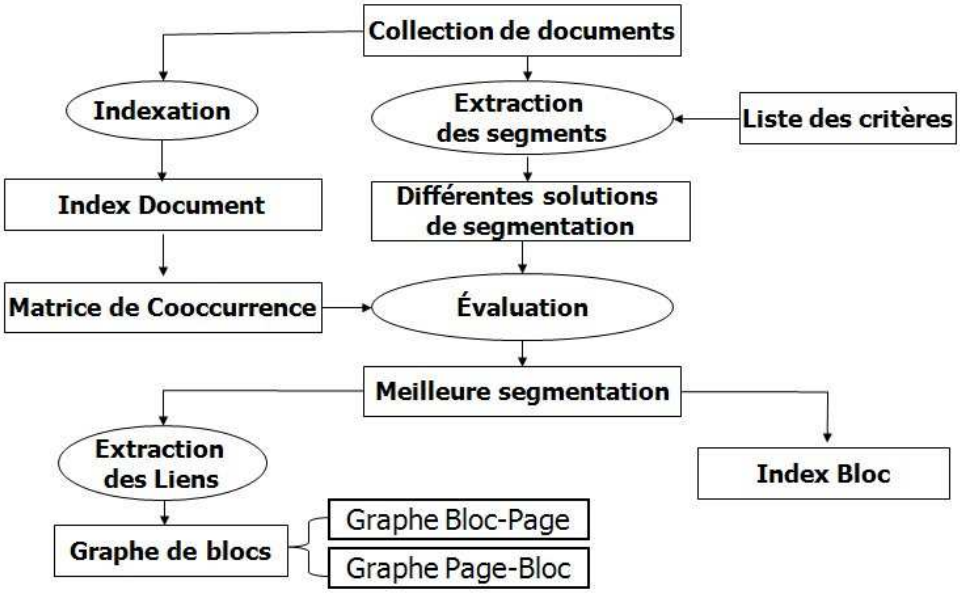


FIGURE 3.8: Processus de segmentation thématique à critères visuels

3.4.3 Processus de segmentation thématique à critères visuels

Afin de segmenter une page Web, on a besoin de déterminer les délimiteurs de blocs. La structure HTML d'une page Web nous offre cette possibilité en utilisant une liste de critères visuels comme les couleurs, les lignes horizontales <HR> et verticales et de la représentation du contenu de la page comme les sous-titres <H1>...<H6>, paragraphes <P> et tables <TABLE>. Le processus de segmentation thématique à critères visuels que nous avons proposé permet de choisir la solution la plus adaptée de telle sorte que les blocs soient cohérents à l'intérieur de leur contenu et distant entre eux. Ce processus fonctionne comme suit :

Pour chaque page de la collection, un index est créé en suivant les étapes d'indexation standard (extraction des mots, lemmatisation et suppression des mots vides). Puis, une matrice de co-occurrence entre termes est déduite à partir de l'index. Cette matrice est utilisée pour évaluer les différentes solutions de segmentation générées à partir de la même page. Ensuite, pour chaque page de la collection, on extrait les différents blocs qui la composent en utilisant les différents délimiteurs de blocs contenus dans la liste des critères. Une solution par critère est générée. Le résultat est un ensemble de solutions de segmentation. Une valeur d'adaptation de notre algorithme génétique de segmentation est calculée pour chacune des solutions. La meilleure solution de segmentation thématique est retenue et l'index bloc est créé. Nous limitons à une seule itération de notre algorithme de segmentation pour des raisons expérimentales. En effet, notre but n'est pas chercher la meilleure segmentation mais de choisir la solution la mieux adaptée à la segmentation de la page et l'utilisée en recherche d'information. De plus, le nombre important de pages à segmenter (plus de 2.5 millions de pages) peut prendre beaucoup de temps d'execution. Et comme il y a pas de méthodes d'evaluation de notre algorithme de segmentation, le recours à plusieurs itérations ne sert à rien à part la perte du temps. Une fois les blocs sont extrait, un graphe de blocs est construit à partir de deux matrices : matrice des liens entre les blocs et les pages pointées par ces blocs et la matrice d'importance entre la page est ces blocs. Les relations page-bloc sont déterminées par l'analyse du contenu de la page. En effet, un bloc est important quand il contient plus d'information de la page. L'importance d'un bloc représente la fraction du nombre de termes du bloc par rapport au nombre de termes de la page. Cependant, les relations bloc-page sont déterminées par l'analyse des liens qui relient les blocs à ces pages. Le but est de construire un graphe bloc-bloc de telle sorte que chaque nœud représente exactement un seul bloc thématique. Les liens entre les blocs sont pondéré en fonction de l'importance de ces

blocs dans les pages. Par exemple, un lien hypertexte qui part d'un bloc vers une page se traduit en plusieurs liens entre le bloc source du lien et les blocs de la page destinataire du lien en pondérant ces liens resultant en fonction de l'importance des blocs de la page destinataire du lien. Une fois l'importance d'un bloc est calculée, cette information serra utilisée dans la fonction de voisinage que nous avons proposée en pondérant les liens en fonction de la valeur β et l'importance du bloc. Ce graphe peut mieux décrire la structure du Web. La figure 3.8 montre les différentes étapes du processus de segmentation thématique à critères visuels que nous avons proposé.

3.4.4 Les inconvénients de la méthode de segmentation

Il est difficile d'utiliser un algorithme génétique sans avoir des contraintes. Il existe des limites qui ne peuvent pas nous ramener à des solutions proche d l'optimum. On peut les résumer en ces quatre points suivants :

1. **Le temps de calcul :** par rapport à d'autres méta-heuristiques, ils nécessitent de nombreux calculs, en particulier au niveau de la fonction d'évaluation. En effet, pour chaque codage binaire d'une solution de segmentation, une construction réel de la segmentation en mémoire à partir de la table des blocs est nécessaire pour calculer la valeur d'adaptation. En plus, en fonction du , les calculs de la cohérence à l'intérieur des blocs et la distance entre les blocs adjacent est proportionnel au nombre de blocs de la solution envisagée. Ceci nécessitent des resources de stockage d'information en mémoire vive ainsi qu'un temps d'exécution un peu élevé. Dans nos experimentation, le temps moyen de segmentation d'une page est de l'ordre de 4 secondes.
2. **Ils sont de plus souvent difficiles à mettre en ouvre :** Des paramètres comme la taille de la population ou le taux de mutation sont parfois difficile à déterminer. Or le succès de l'évolution en dépend et plusieurs essais sont donc nécessaires, ce qui limite encore l'efficacité de l'algorithme. En outre, choisir une bonne fonction d'évaluation est aussi critique. Celle-ci doit prendre en compte les bons paramètres du problème. Elle doit donc être choisit avec soin. Nous ne savons pas si notre fonction d'adaptation est bonne ou non faute de l'inexistante de méthodes d'evaluation de différentes méthodes de segmentation. Cependant, nous avons évalué l'impact des blocs resultant de l'application de notre algorithme de segmentation dans la recherche d'information. C'est le seul moyen à nos yeux afin de pouvoir évaluer l'algorithme génétique de segmentation que nous avons proposé.
3. **Il faut aussi noter l'impossibilité d'être assuré :** Même après un nombre important de générations, que la solution trouvée est la meilleure. On peut seulement être sûr que l'on s'est approché de la solution optimale sans la certitude de l'avoir atteinte.
4. **optimums locaux :** Un autre problème important est celui des optimums locaux. En effet, lorsqu'une population évolue, il se peut que certaines solutions de segmentation qui à un instant occupent une place importante au sein de cette population deviennent majoritaires. À ce moment, il se peut que la population converge vers cette solution et s'écarte ainsi de la solution de segmentation la plus intéressants.

3.4.5 Complexité de l'algorithme de segmentation

La complexité d'un algorithme correspond au nombre d'opérations élémentaires (affectations, comparaisons, opérations arithmétiques) effectuées par l'algorithme. Elle s'exprime en fonction de la taille n des données. On dit que la complexité de l'algorithme est $O(f(n))$ où f est d'habitude une combinaison de polynômes, logarithmes ou exponentielles. Ceci reprend la notation mathématique classique, et signifie que le nombre d'opérations effectuées est borné par $c * f(n)$, où c est une constante, lorsque n tend vers l'infini. Considérer le comportement à l'infini de la complexité est justifié par le fait que les données des algorithmes sont de

grande taille et qu'on se préoccupe surtout de la croissance de cette complexité en fonction de la taille des données. Dans ce qui suit, nous allons détaillé la complexité de chaque fonction utilisée dans le processus de segmentation des pages et la complexité de notre algorithme génétique.

3.4.5.1 Fonction de la co-occurrence

Cette fonction calcule la co-occurrence entre les termes de la collection. t_c représente le nombre de termes de la collection. Le tri de la liste des termes, nous permet une recherche rapide de la co-occurrence de deux termes quelconques. La complexité d'une telle fonction est la suivante :

$$O(ocur) = O\left(\underbrace{t_c * \log(t_c)}_{(1)} + \underbrace{\frac{t_c^2 + t_c}{2}}_{(2)}\right) = O(n^2)$$

- (1) : Tri des termes de la collection
- (2) : Deux boucles pour calculer la co-occurrence entre chaque deux termes de la collection

3.4.5.2 Cohérence entre deux blocs

Cette fonction permet de calculer la cohérence à l'intérieur d'un bloc. t_b représente le nombre de termes d'un bloc quelconque. La complexité de cette fonction dépend de deux paramètres le nombre de termes de la collection t_c et le nombre de termes d'un blocs t_b. Au pire des cas la complexité de cette fonction est de $O(N^4 * (log(N)))$.

$$O(coh) = O\left(\underbrace{\frac{t_b^2 + t_b}{2}}_{(2)} * \left(\underbrace{\left(\frac{t_c^2 + t_c}{2}\right) * \log\left(\frac{t_c^2 + t_c}{1}\right)}_{(2)}\right)\right) = O\left(n^4 \log(n)\right)$$

- (1) : Deux boucles pour calculer la somme de la co-occurrence entre les termes d'un même bloc
- (2) : Recherche de la co-occurrence entre deux termes dans un tableau trié.

3.4.5.3 Distance entre deux blocs adjacents

La distance entre deux blocs est calculée en fonction de deux mesures. La première mesure représente la cohérence entre deux blocs et la deuxième mesure la similarité entre deux vecteurs blocs en terme du nombre de termes en communs. Comme la cohérence dans un bloc, cette fonction dépend de deux paramètres le nombre de termes de la collection t_c et le nombre de termes d'un bloc t_b. La complexité de la fonction est calculée comme suit :

$$O(dist) = O\left(\underbrace{\underbrace{t_{b1} * t_{b2}}_{(1)} * \underbrace{\left(\frac{t_c^2 + t_c}{2}\right) * \log\left(\frac{t_c^2 + t_c}{2}\right)}_{(2)}}_{(3)} + \underbrace{\underbrace{t_{b1} + t_{b2}}_{(4)} + \underbrace{t_{b1} * t_{b2}}_{(5)}}_{(6)}\right) = O\left(n^4 \log(n)\right)$$

- (1) : Deux boucles pour calculer la somme de la co-occurrence entre les termes du bloc 1 et les termes du bloc 2
- (2) : Recherche de la co-occurrence entre deux termes dans un tableau trié
- (3) : Calcul de la coherence entre deux blocs adjacents
- (4) : Calcul de la somme carrée des fréquences des termes de chaque bloc.

– (5) : Deux boucles pour le calcul du produit des fréquences entre les termes du blocs 1 et les termes du bloc 2
– (6) : Calcul de la similarité entre deux blocs

3.4.5.4 Fonction d'evaluation d'une solution de segmentation d'une page

La fonction d'evaluation d'une solution de segmentation d'une seule page dépend du nombre de blocs de la solution n_b et de la complexité des deux fonctions cohérence à l'intérieur d'un bloc et la distance entre des blocs adjacents.

$$O\left(evalS\right) = O\left(\underbrace{n_b}_{(1)} + \underbrace{n_b * O\left(coh\right)}_{(2)} + \underbrace{\left(n_b - 1\right) * O\left(dist\right)}_{(3)}\right) = O\left(n^5 \log\left(n\right)\right)$$

– (1) : Découpage d'une page en plusieurs blocs selon des critères visuels
– (2) : Calcul de la coherence d'un bloc
– (3) : Calcul de la distance entre deux blocs adjacents

3.4.5.5 Segmentation des pages d'une collection

L'algorithme de segmentation que nous avons utilisé a une complexité élevée. Cette complexité dépend du nombre de pages considérées n_p, du nombre de délimiteurs de blocs d_s et enfin de la complexité de la fonction d'evaluation d'une solution de segmentation. Ce calcul ne tient pas compte ni des opérateurs, ni du nombre d'itération de notre algorithme génétique de segmentation. La complexité de cette fonction est définie comme suit :

$$O\left(SegP\right) = O\left(n_p * \left(\underbrace{\left(d_s * n_b\right)}_{(1)} + d_s * O\left(evalS\right)\right)\right) = O\left(n^7 \log\left(n\right)\right)$$

– (1) : Découpage d'une page en plusieurs blocs selon des critères visuels

La complexité de notre algorithme est polynomial d'ordre 9. Il n'est pas adapté à la segmentation des pages du Web entiere mais plutôt pour des corpus de documents de petite taille. En effet, l'algorithme devient lent avec l'augmentation du nombre de page, du nombre d'itération et/ou de la taille de chaque population.

3.4.5.6 Algorithme génétique de segmentation

Nous avons aussi calculé la complexité de notre algorithme génétique de segmentation de page Web en tenant compte de toutes les étapes du processus de segmentation. Nous remarquons que la complexité de cet algorithme génétique de segmentation dépend du nombre d'itération de l'algorithme n_i nécessaire afin de retrouver une solution proche de l'optimum, du nombre de pages à ségmenter n_p, du nombre de solutions de segmentation d'une page contenus dans chaque population intermédiaire n_s et du nombre de blocs par page n_b. Elle est calculée comme suit :

$$O\left(SegG\right) = O\left(n_p * n_i * \left(\underbrace{\left(n_s * n_b\right)}_{(1)} + \underbrace{n_s}_{(2)} + \underbrace{\frac{n_s}{2}}_{(3)} + \underbrace{n_s * O\left(evalS\right)}_{(4)}\right)\right) = O\left(n^9 \log\left(n\right)\right)$$

– (1) : Découpage d'une page en plusieurs blocs selon des critères visuels
– (2) : La mutation d'une solution de segmentation
– (3) : Le croisement entre deux solutions de segmentation
– (4) : Evaluation de la nouvelle generation

3.5 Conclusion

Dans ce chapitre nous avons proposé un nouveau modèle de propagation de pertinence reposant sur la pondération dynamique des liens entre les différents unités d'information (bloc, page et site). Nous avons aussi présenté une architecture de notre modèle en trois couches (bloc, page et site), ainsi que les différents modules qui le composent. De plus, nous avons décrit notre algorithme génétique de segmentation de pages Web en blocs thématiques reposant sur les critères visuels et les critères de representation du contenu des pages HTML. Cet algorithme consiste à évaluer plusieurs solutions de segmentation et de choisir parmi elles la meilleure solution adaptée à notre problème de segmentation. Nous avons fait une analogie entre le problème de segmentation des pages Web et les algorithmes génétiques et nous avons proposé une fonction d'évaluation d'une solution de segmentation reposant sur la coherence du contenu textuel à l'intérieur du bloc et des distances entre les blocs adjacents d'une segmentation. Nous avons aussi détaillé chaque étape de notre algorithme génétique de segmentation et les limites qui découlent de cet algorithme. Notre modèle répond aux différents problèmes posés par les techniques d'analyse de liens que nous avons étudiées dans la partie bibliographique de notre thèse 2. Nous avons essayé de répondre au problème de spamming des liens par notre modèle de propagation de pertinence. En effet, avec notre modèle, il ne suffit pas d'ajouter des liens entrants à un document pour qu'il figure au top du classent. Il faut aussi ajouter les termes autour des liens pour que le document aura un poids important pour une requête spécifique. De ce fait, on passe de spamming de liens vers le spamming de termes. Or, il est plus difficile d'ajouter des liens avec beaucoup de termes au tour de ce lien dans les documents qui jouent un rôle important dans le graphe du Web et qui pointent vers un document spam. On parle du spamming dépendant de termes de la requête. En effet, l'ajout de termes au texte ancre des liens et au tour de ces liens va certainement augmenter le score de pertinence de ces documents par rapport aux requêtes qui contiennent ces termes, mais pas par rapport à toutes les requêtes comme le cas des techniques d'analyse de liens étudiées. L'amélioration du classement d'un document nécessite de cibler un type de requêtes, mais pas toutes les requêtes.

Le deuxième problème concerne le sens des thèmes. Avec l'algorithme de segmentation que nous avons proposé, les pages Web peuvent être découper en blocs de thématiques différentes. La différence de notre méthode de segmentation par rapport aux autres méthodes structurelles existantes de découpage de page Web reside dans le choix de la meilleure segmentation en comparant différentes solutions de segmentation. Avec notre algorithme de segmentation, on peut avoir des segmentation à plusieurs critères de delimitations de blocs, des segmentation séquentielles et même hiérarchiques.

Dans le chapitre suivant, nous présentons les différentes expérimentations effectuées sur les deux collections issues de la conférence TREC qui sont WT10g et Gov.

Chapitre 4

Expérimentations sur les collections TREC et GOV

Dans le cadre de nos expérimentations, nous avons choisi comme collections de tests la collection WT10g issue du corpus de la conférence TREC-10 ayant eu lieu en 2001 et la collection GOV issue du corpus de la conférence TREC-11 ayant eu lieu en 2002. Nous les avons choisies en raison de la notoriété des collections issues de la compagne TREC et par conséquent, leur statuts de collections standards dans le domaine de recherche d'information.

Les deux collections de test WT10g et GOV sont définies par un ensemble de pages, un ensemble de 50 requêtes et un ensemble de jugements de pertinence par rapport à chaque requête. La collection WT10g est composée de pages de l'index du moteur de recherche Alta vista de l'année 1997, tandis que la collection GOV est composée essentiellement des pages du gouvernement américain (domaine .gov). Chaque requête est composée de trois champs : le titre, la description et la narration de la requête. Le titre correspond à la requête saisie par l'utilisateur. Cependant, la description et la narration de la requête permet de décrire en détail la recherche demandée et la nature des documents qui sont pertinents à la requête et ceux qui ne le sont pas.On distingue deux ensembles de 50 requêtes :

- Topic2001 : un ensemble de 50 requêtes exécutées sur la collection WT10g. Chaque requête est composée en moyenne de 2.7 termes.
- Topic2002 : un ensemble de 50 requêtes exécutées sur la collection GOV. Chaque requête est composée en moyenne de 2.92 termes.

Nous voulons connaître à partir de ces expérimentations l'impact de notre fonction de voisinage sur les différents niveaux de l'architecture de notre système. Tout d'abord, une comparaison entre les deux collections s'impose afin de déterminer les caractéristiques de chacune d'elles.

Nous les avons comparées par rapport au nombre de documents qu'elles contiennent et la connectivité du graphe de liens généré à partir les liens hypertextes qui relient les pages de la même collection. Le tableau 4.1 montre les caractéristiques de chacune des deux collections.

Nous remarquons que la collection WT10g est moins volumineuse que la collection GOV. Les pages de la collection GOV contiennent plus de termes que celles de la collection WT10g. En effet, la taille moyenne d'une page dans la collection WT10g est de 127 termes, alors que la taille moyenne d'une page dans la collection GOV est de 171 termes. Nous remarquons aussi que la connectivité du graphe de liens de la collection GOV est relativement dense par rapport au graphe du liens de la collection WT10g. Et comme la collection WT10g est un échantillon du Web, Il est fort possible que les liens hypertextes qui relient les pages de la collection GOV soient des liens de qualité par rapport ceux qui relient les pages de la collection WT10g. En effet, dans le Web,

	WT10g		GOV	
Nombre de documents	1692096		1247753	
Nombre de documents avec des liens entrants	1295841	76.5%	1067339	85.5%
Nombre de documents avec des liens sortants	1532012	90.5%	1146213	91.6%
Nombre moyen de liens entrants/page	5.26		10.4	
Nombre moyen de liens sortants/page	6.22		9.69	

TABLE 4.1: Caractéristiques des collections de tests WT10g et GOV

n'importe qui peut citer n'importe quoi dans ces pages, alors que dans le domaine .gov, les liens de citations des loi du gouvernement américain doivent être valides et renforcés par l'existence d'une complémentarité entre les documents reliés par les liens hypertextes.

4.1 Métriques d'évaluation

Dans la première partie des expérimentations, nous avons considéré la page Web comme étant la plus petite unité d'information à retourner à l'utilisateur. Nous avons comparé la fonction de correspondance de notre système ($S_p(P,Q)$) par rapport à d'autres fonctions de correspondance existantes : le contenu seul ($OKAPI_D(P,Q)$) , la popularité (PageRank et HITS) et la propagation de pertinence reposant sur un paramètre de propagation statique (RSV Relevance Status Value). Notre fonction de correspondance n'est autre que la propagation de pertinence reposant sur un paramètre de propagation dynamique calculé en fonction du nombre de termes de la requête contenus dans les pages Web répondant à la requête utilisateur.

Dans la deuxième partie des expérimentations, nous avons considéré le bloc thématique comme la plus petite unité d'information retournée par un moteur de recherche. Nous avons comparé notre fonction de correspondance reposant sur le le contenu seul des blocs générés en utilisant un algorithme génétique de découpage de pages en blocs thématiques ($OKAP_B$), cet algorithme repose sur l'évaluation de plusieurs solutions de segmentation d'une même page, par rapport au contenu seul des pages ($OKAPI_D$), contenu seul des blocs séparés par les balises lignes horizontales <HR> ($OKAPI_H R$), le contenu seul des blocs séparés par les balises headings <H1> au <H6> ($OKAPI_H$), le contenu seul des blocs séparés par les balises retour à la ligne
 ($OKAPI_B R$) et le contenu seul des blocs séparés par les balises paragraphes <P> ($OKAPI_P$). Le but d'une telle comparaison est de voir l'impact de notre algorithme de découpage de pages Web que nous avons proposé dans le chapitre 3 sur la precision de la recherche effectuée. Notre algorithme de segmentation repose sur plusieurs critères de segmentation tels que headings <H1> au <H6>, lignes horizontales <HR>, retour à la ligne <BR> et paragraphes <P>.

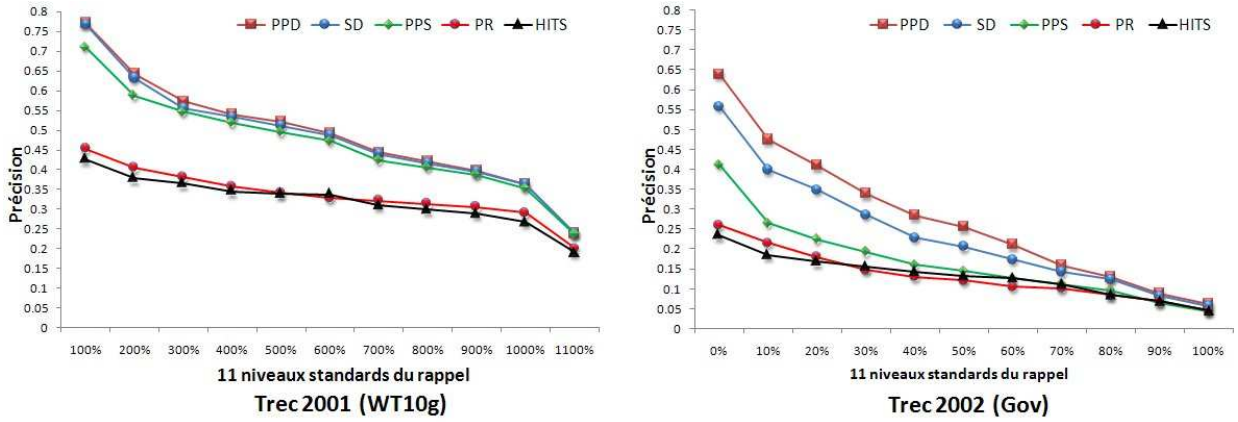
Dans la troisième partie, nous avons comparé plusieurs variantes de notre fonction de correspondance au niveau bloc, page et site en tenant compte de la propagation de pertinence à travers le graphe des blocs, le graphe des pages et le graphe des sites.

La mesure principale d'évaluation de nos expérimentations est la précision moyenne aux 11 niveaux standard du rappel qui sont 0%, 10%, 20%,..,100% du rappel. Nous avons aussi évalué la précision moyenne globale MAP et la précision obtenue à 5 et à 10 documents retrouvés (P@5, P@10) ainsi que le succès au 1er document retrouvé (S@1). De plus, nous avons évalué le gain de la precision en MAP, P5 et P10 pour chaque requête exécutée. Cette dernière permet de comparer deux fonctions de correspondance par rapport chaque requête.

4.2 Expérimentations au niveau page

Dans cette section, nous présentons les différentes expérimentations réalisées au niveau page en comparant notre système par rapport au contenu seul $OKAPI_D$ et les techniques standards d'analyse de liens (PageRank, HITS et RSV) selon plusieurs critères d'évaluation.

Dans un premier lieu, nous comparons les différentes fonctions de correspondance par rapport à la précision moyenne aux 11 niveaux standard du rappel.



(a) (b)

FIGURE 4.1: La précision moyenne aux 11 niveaux standards du rappel pour les différentes fonctions de correspondance utilisées

La figure 4.1 montre les résultants expérimentaux obtenus sur les deux collections WT10g et GOV en utilisant cinq fonctions de correspondance. La première fonction noté $OKAPI_D(P, Q)$ (score de pertinence de la page P par rapport à la requête utilisateur Q reposant sur le contenu seul de la page). Cette fonction représente l'algorithme de base de nos évaluations. La deuxième fonction repose sur la combinaison du contenu textuel des pages et leur voisinage immédiat. Cette fonction noté $S_p(P, Q)$ (propagation de pertinence reposant sur un paramètre de propagation dynamique) s'appuie sur la propagation de pertinence entre les pages à travers le graphe des liens dont la pondération de ces liens est dynamique et proportionnelle au nombre de termes de la requête contenus dans ces pages. La troisième fonction est une propagation de pertinence reposant sur un paramètre de propagation statique (PPS) dont la pondération des liens est fixée à priori. Le principe de cette fonction réside dans la propagation d'une portion du score de pertinence de la page source d'un lien vers la page destinataire dur lien. La quatrième et la cinquième fonction reposent sur la popularité de la page. Pour cela, nous avons utilisé les deux algorithmes les plus connus des algorithmes d'analyse des liens qui sont le PageRank noté (PR) et HITS. La différence entre ces deux algorithmes est que le PageRank calcule une valeur de popularité de chaque page de la collection en utilisant le graphe des liens en entier, tandis que l'algorithme HITS s'applique à un espace de travail composé de l'ensemble de pages de la collection en réponse à la requête utilisateur. Ces différents algorithmes sont décrits dans le tableau 4.2.

D'après la figure 4.1, on constate que les deux algorithmes d'analyse des liens PageRank et HITS réalisent les plus mauvais résultats par rapport aux trois autres algorithmes sur les deux collections TREC et GOV. Avec ces deux algorithmes, une page aura un score de popularité indépendant des scores de pertinence de son voisinage pour chaque requête exécutée sur le système. C'est pour cette raison que les résultats obtenus sont mauvais.

La combinaison entre une mesure reposant sur le contenu d'une page et le voisinage de cette page qui représente le score de propagation de pertinence en fixant le paramètre propagation n'améliore pas les résultats

Algorithmes	Fonctions
SD	$Okapi(P,Q)$
PPD	$Okapi(P,Q) + VD(P,Q)$ où $VD(P,Q)$ représente le voisinage des pages que nous avons proposé et qui est calculé en fonction du nombre de termes de la requête contenu dans les pages qui pointent vers P.
PPS	$\mathrm{Okapi(P,Q)} + \sum_{P' \rightarrow P} 0.25 * \mathrm{Okapi(P,Q)}$
PR	$Okapi(P,Q) + PageRank(P)$
HITS	$Okapi(P,Q) + (Hub(P,Q) + Authority(P,Q))/2$

TABLE 4.2: Les différentes fonctions de correspondance exécutées sur les deux collection WT10g et GOV

obtenus sur les deux collections TREC et GOV avec la fonction du contenu seul. Au contraire, les résultats obtenus avec cette fonction sont moins bons par rapport à l'algorithme de base. La chute de la précision peut être justifiée par l'introduction du bruit dans le calcul des scores de pertinence des documents lors de la propagation de pertinence. En effet, ces systèmes traitent tous les liens hypertextes de la même façon (pondération des liens en utilisant le même paramètre de propagation). De plus, ces systèmes ne distinguent pas entre les liens porteurs des informations entières de la recherche et les liens porteurs des informations partielles de la recherche demandée ainsi que les liens vides qui n'apportent pas d'informations additionnelles à la recherche.

Enfin, la combinaison d'une mesure reposant sur le contenu d'une page et de son voisinage immédiat dont la propagation de pertinence entre les pages repose sur la pondération dynamique des liens entre ces pages donne de meilleurs résultats par rapport à l'algorithme de base. Ceci signifie que la combinaison d'une mesure du contenu d'une page et de son voisinage calculé dynamiquement en fonction des termes de la requête peut apporter plus de précision dans les résultats retournés par un moteur de recherche classique. Cependant, l'augmentation de la precision au 11 niveaux standards du rappel pour la collection GOV est plus important que celui de la collection WT10g. En effet, l'augmentation de la precision varie entre 1% et 3% pour la collection WT10g et entre 5% et 25% pour la collection GOV. Cela est justifié par la qualité des liens qui relient les pages des deux collections et la densité de ces liens. Comme la collection GOV est une collection de documents gouvernementaux, les liens qui relient ces documents sont considérés comme des liens informationnels qui apportent de l'information additionnelle à la page destinataire du lien. Alors que dans la collection WT10g, les documents viennent du Web. D'où l'existence des liens spams ou des liens vides qui n'apportent pas de nouvelles informations à la page destinataire du lien et qui introduisent du bruit dans le calcul du score de pertinence de ces pages. De plus, la densité des liens dans la collection GOV est plus importante que dans la collection WT10g. la dépendance entre le paramètre de combinaison α et la précision moyenne MAP, P@5 et P@10 pour les systèmes de propagation de pertinence (statique et dynamique) exécutés sur les deux collections WT10g et GOV est illustré dans la figure 4.2. L'observation de ces courbes permet de constater que :

- Les résultats obtenus par le système reposant sur la propagation de pertinence statique se dégradent rapidement en fonction du paramètre de propagation fixe α. En effet, en donnant plus d'importance aux scores du voisinage des pages, la precision MAP, P5 et P10 diminuent considérablement sur les deux collections WT10g et GOV.

- Les résultats obtenus par le système reposant sur la propagation de pertinence dynamique en le combinant avec le contenu seul de la page réalisent des performances meilleures que celles reposant sur le contenu seul de la page. Ces résultats reflètent le bon fonctionnement de notre système sur les deux collections WT10g et GOV.

Nous remarquons aussi que les meilleurs résultats obtenus avec notre système montrent un gain global moyen de 2%, 8% et 6% par rapport à l'algorithme de base (système reposant sur le contenu seul des pages) sur la

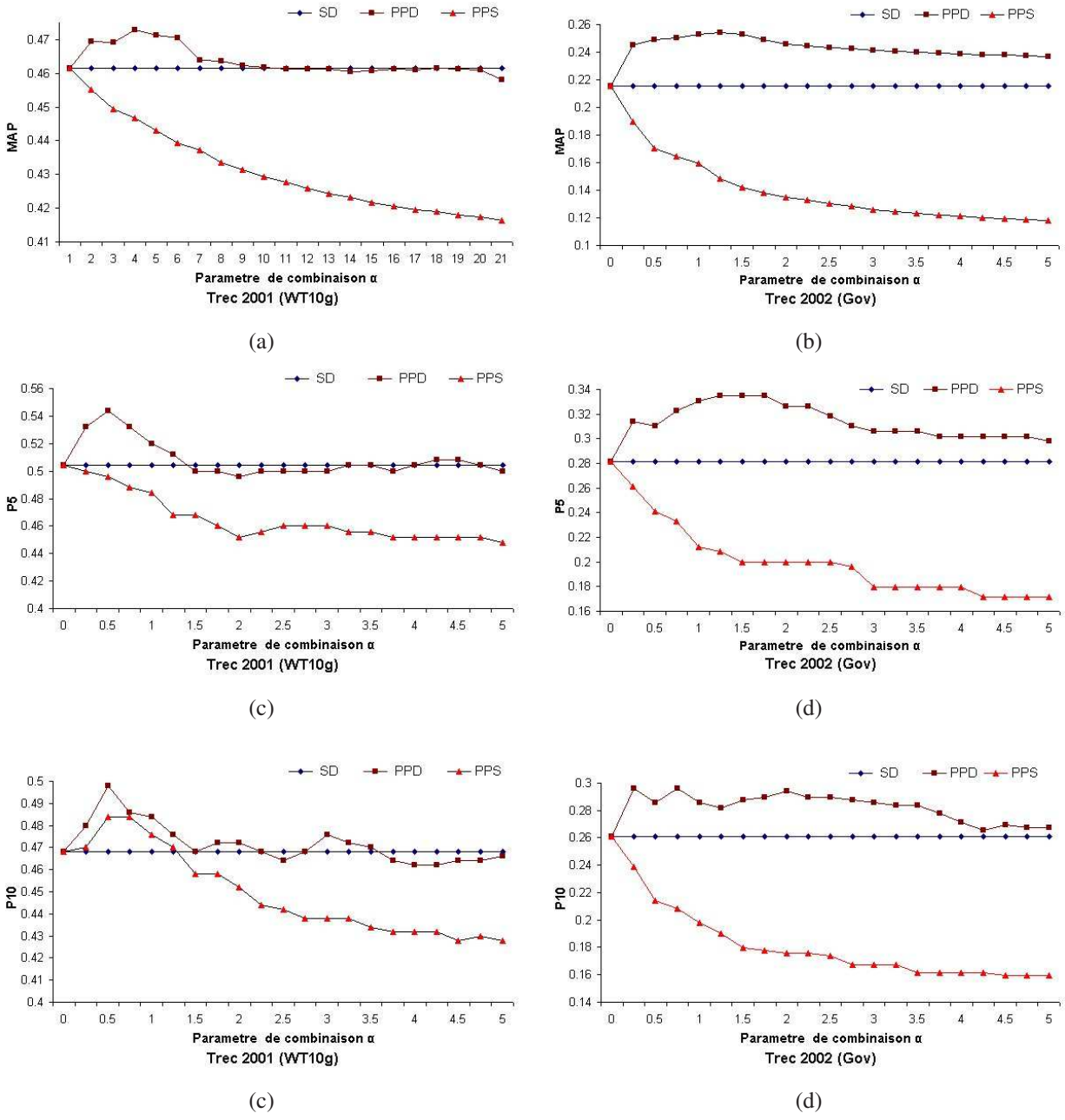


FIGURE 4.2: Comparaison entre différentes fonctions de correspondance en fonction du paramètre de combinaison α

précision moyenne MAP, P5 et P10 respectivement pour la collection WT10g (0.46, 0.5 et 0.47 respectivement pour l'algorithme de base). Le gain global moyen pour la collection GOV est de 18%, 19% et 13% par rapport à l'algorithme de base sur la précision moyenne MAP, P5 et P10 respectivement (0.21, 0.28 et 0.26 respectivement pour l'algorithme de base). Les résultats obtenus avec la collection GOV sont meilleurs que ceux obtenus avec la collection WT10g. Afin de voir d'où vient ce gain important, une étude détaillée du gain de la precision en MAP, P5 et P10 pour chaque requête exécutée sur notre système est indispensable. La figure 4.3 représente le gain de la precision en MAP, P5 et P10 obtenu pour chaque requête exécutée avec notre système sur les deux collections WT10g et GOV. Les histogrammes montrent un comportement similaire sur les mesures d'évaluation de la precision moyenne MAP, P5 et P10. Les améliorations sont généralement bien plus grandes et bien plus fréquentes que les dégradations. En regardant les requêtes individuellement, nous remarquons que les résultats obtenus par 52% des requêtes environ montrent des améliorations significatives de la precision MAP, P5 et P10 sur les deux collections WT10g et GOV. Les résultats obtenus par 23% environ des requêtes montrent peu d'amélioration sur les deux collections et pour environ 25% des requêtes, tenir compte du voisinage des pages dans le calcul du score de pertinence des pages dégradent la precision moyenne MAP, P5 et P10 sur les deux collection WT10g et GOV.

L'analyse des requêtes pour lesquelles notre système obtient de mauvais résultats montre que ces requêtes cherchent des réponses qui se trouvent dans le corps de la page (exemple "do beavers live in salt water ?") alors que les requêtes pour lesquelles notre système obtient de bons résultats cherchent des réponses qui se trouvent au début de la page. En effet, les termes du texte ancre et autour des liens hypertextes décrivent d'une manière générale le contenu global de la page cible du lien. Ces termes sont repris dans les titres et les introductions des pages. Les auteurs des liens hypertextes utilisent ces termes afin de décrire la page cible (exemple "jennifer aniston").

Le tableau 4.3 montre les résultats obtenus sur les deux collections TREC et GOV en comparant différentes variantes de notre système sur les deux mesures d'evaluation MAP et P5. Le premier système est celui qui tient compte de l'importance des termes dans le calcul du score de pertinence de la page. L'importance des termes est obtenu en propageant les poids des termes se trouvant au tour des liens hypertextes à travers le graphe des liens. Nous avons détaillé le processus de propagation des poids des termes dans la partie modèle de notre système. Le deuxième système est celui qui combine le contenu seul de la page et de son voisinage calculé dynamiquement en propageant les scores de pertinence des pages voisines à travers le graphe des liens. Enfin, le dernier système qui combine en plus du contenu seul de la page et du voisinage immédiat des pages, le voisinage du site contenant la page. Nous avons comparé les différentes combinaisons de ces trois algorithmes par rapport à l'algorithme de base. Les résultats montrent des améliorations de nos systèmes par rapport à l'algorithme de base. Le meilleur système est celui qui combine le contenu seul des pages, le voisinage immédiat de ces pages et l'importance des termes. Cependant, les résultats obtenus en combinant le contenu seul, le voisinage immédiat des pages et le voisinage des sites sont un peu au dessous du système de propagation de pertinence dynamique PDD. Ce qui veut dire que l'information additionnelle provenant du site pour la requête n'est pas importante. Ceci est justifié par la multitude de thèmes abordés dans un site et la diversité du contenu des pages d'un site. On retrouve des pages d'ordre personnel, commercial ou informationnel qui touchent des domaines d'activités différentes. L'importance des termes dans le calcul des poids de termes, elle aussi produit peu d'amélioration par rapport à l'algorithme de base. En effet, il est peu fréquent que les requêtes exécutées sur notre système contiennent des termes figurant dans le texte ancre et autour des liens hypertextes, d'où une augmentation minime par rapport à l'algorithme de base. Une combinaison du contenu seul, du voisinage immédiat des pages et de l'importance des termes dans le calcul des poids des termes s'avère utile pour la recherche générique sur le Web.

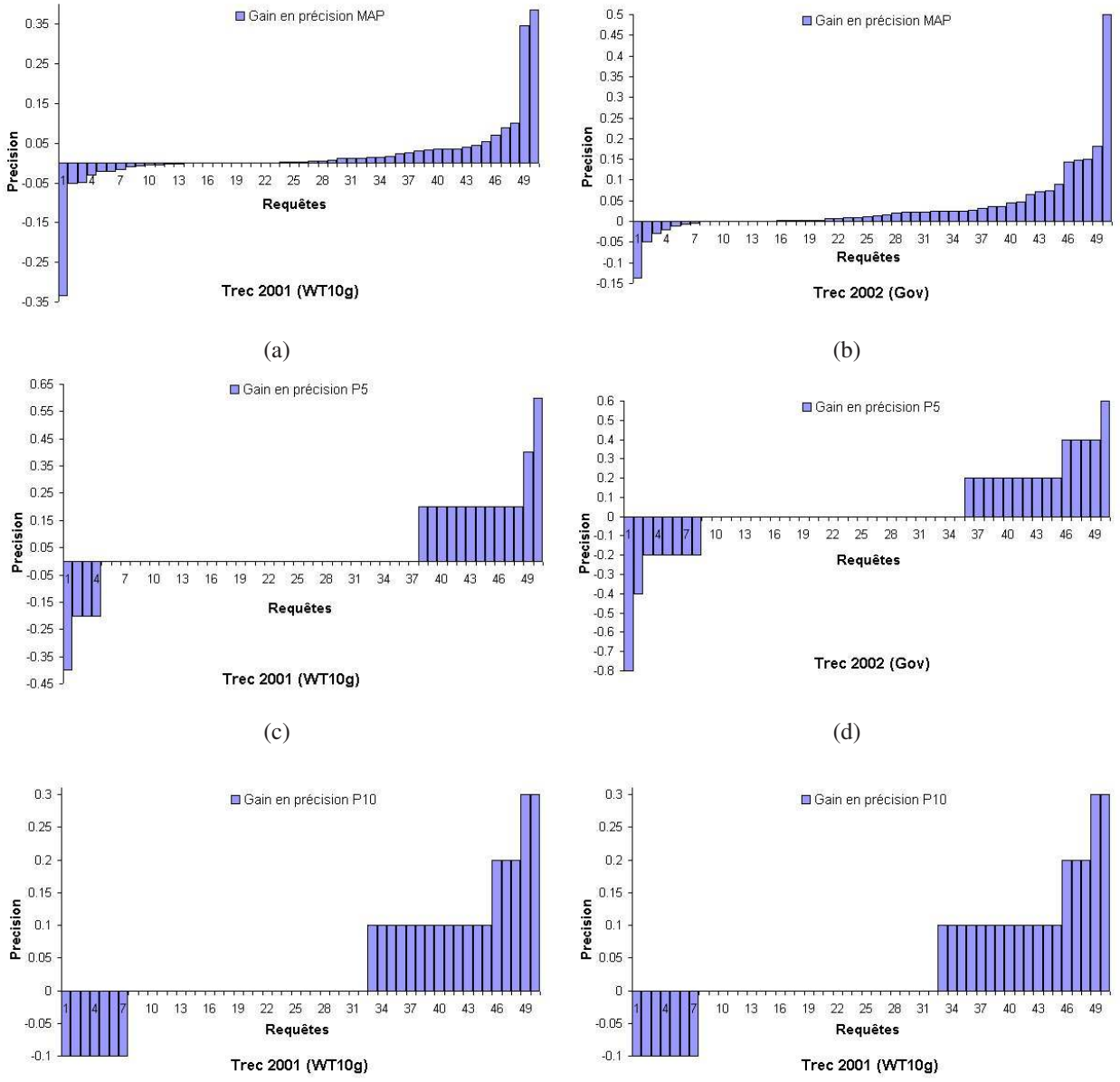


FIGURE 4.3: Gain de la précision en MAP, P5 et P10 de la propagation de pertinence dynamique PPD par rapport au contenu seul SD

α	MAP											
	Trec 2001 (WT10g)						Trec 2002 (GOV)					
	Sans PPT			Avec PPT			Sans PPT			Avec PPT		
	SD	**PPD**	**PPD+VS**	**SDimp**	**PPDimp**	**PPDimp +VSimp**	**SD**	**PPD**	**PPD+VS**	**SDimp**	**PPDimp**	**PPDimp +VSimp**
0	46.1%	46.1%	46.1%	46.8%	46.8%	46.8%	21.5%	21.5%	21.5%	21.7%	21.7%	21.7%
0.25	46.1%	46.94%	46.71%	46.8%	47.3%	46.91%	21.5%	23.75%	22.4%	21.7%	24.51%	22.81%
0.5	46.1%	46.92%	47.08%	46.8%	46.99%	47.14%	21.5%	24.44%	22.16%	21.7%	24.93%	21.47%
0.75	46.1%	47.13%	47.02%	46.8%	47.17%	46.78%	21.5%	24.41%	21.66%	21.7%	25.03%	20.75%
1	46.1%	47.06%	46.77%	46.8%	46.98%	46.47%	21.5%	24.82%	21.51%	21.7%	25.3%	20.49%
1.25	46.1%	47.06%	46.69%	46.8%	46.4%	46.35%	21.5%	24.87%	21.34%	21.7%	25.39%	20.33%
1.5	46.1%	46.38%	46.55%	46.8%	46.36%	46.21%	21.5%	24.97%	20.84%	21.7%	25.31%	20.12%
1.75	46.1%	46.36%	46.32%	46.8%	46.09%	46.02%	21.5%	24.88%	20.63%	21.7%	24.89%	19.92%
2	46.1%	46.24%	46.3%	46.8%	46.06%	45.87%	21.5%	24.58%	20.41%	21.7%	24.57%	19.7%
MAX	**46.1%**	**47.13%**	**47.08%**	**46.8%**	**47.3%**	**47.14%**	**21.5%**	**24.97%**	**22.4%**	**21.7%**	**25.39%**	**22.81%**

α	P5											
	Trec 2001 (WT10g)						Trec 2002 (GOV)					
	Sans PPT			Avec PPT			Sans PPT			Avec PPT		
	SD	**PPD**	**PPD+VS**	**SDimp**	**PPDimp**	**PPDimp +VSimp**	**SD**	**PPD**	**PPD+VS**	**SDimp**	**PPDimp**	**PPDimp +VSimp**
0	50.4%	50.4%	50.4%	51.2%	51.2%	51.2%	28.16%	28.16%	28.16%	29.39%	29.39%	29.39%
0.25	50.4%	53.2%	52.8%	51.2%	52.4%	52%	28.16%	31.43%	29.8%	29.39%	30.61%	29.76%
0.5	50.4%	53.6%	52.4%	51.2%	54%	53.6%	28.16%	31.02%	28.16%	29.39%	31.02%	27.35%
0.75	50.4%	53.2%	52.8%	51.2%	53.2%	51.6%	28.16%	32.24%	25.71%	29.39%	32.24%	25.31%
1	50.4%	52%	50.4%	51.2%	51.2%	48%	28.16%	33.06%	26.12%	29.39%	32.24%	24.49%
1.25	50.4%	51.2%	49.2%	51.2%	49.2%	48.8%	28.16%	33.47%	25.31%	29.39%	33.65%	23.67%
1.5	50.4%	50%	49.6%	51.2%	49.2%	48.4%	28.16%	33.47%	26.53%	29.39%	33.47%	24.9%
1.75	50.4%	50%	47.6%	51.2%	48%	48%	28.16%	33.47%	26.94%	29.39%	33.06%	24.49%
2	50.4%	49.6%	47.6%	51.2%	48.8%	47.6%	28.16%	32.65%	26.12%	29.39%	32.55%	24.49%
MAX	**50.4%**	**53.6%**	**52.8%**	**51.2%**	**54%**	**53.6%**	**28.16%**	**33.47%**	**29.8%**	**29.39%**	**33.65%**	**29.76%**

TABLE 4.3: Comparaison entre différentes systèmes de recherche tenant compte des liens dans la fonction de correspondance au niveau page

Enfin, en ce qui concerne la rapidité de retrouver les documents pertinents, illustré dans le tableau 4.4, Notre système reste plus performant que les autres systèmes. En effet, le nombre de requête de notre système dont la premiere page retrouvée est pertinente à la recherche enregistre une augmentation de 18,5% et 19% par rapport à l'algorithme de base sur les deux collections WT10g et GOV respectivement. La propagation de pertinence reposant sur un paramètre de propagation statique, le page rank et l'algorithme HITS restent toujours au dessous de l'algorithme de base et de notre système. Le même constat est dressé par la mesure de succès à 5 et 10 documents retrouvés (S@5 et S@10). Cette observation montre qu'il est fort possible que la premiere page retournée par notre système soit pertinente à la recherche effectuée.

	Trec 2001(WT10g)								
	SD	**PPD**		**PPS**		**PR**		**HITS**	
S@1	27	32	18.5%	21	-22.2%	11	-59.3%	10	-63%
S@5	42	44	4.8%	42	0%	27	-35.7%	22	-47.6%
S@10	46	48	4.3%	46	0%	33	-28.3%	30	-34.8%
	Trec 2002(GOV)								
	SD	**PPD**		**PPS**		**PR**		**HITS**	
S@1	21	25	19%	12	-42.9%	5	-76.2%	5	-76.2%
S@5	30	37	23.3%	25	-16.7%	15	-50%	14	-53.3%
S@10	36	40	11.1%	33	-8.3%	23	-36.1%	20	-44.4%

TABLE 4.4: Comparaison entre différentes fonctions de correspondance par rapport au succès au 1^{er},5^{eme} et 10^{eme} documents retrouvés

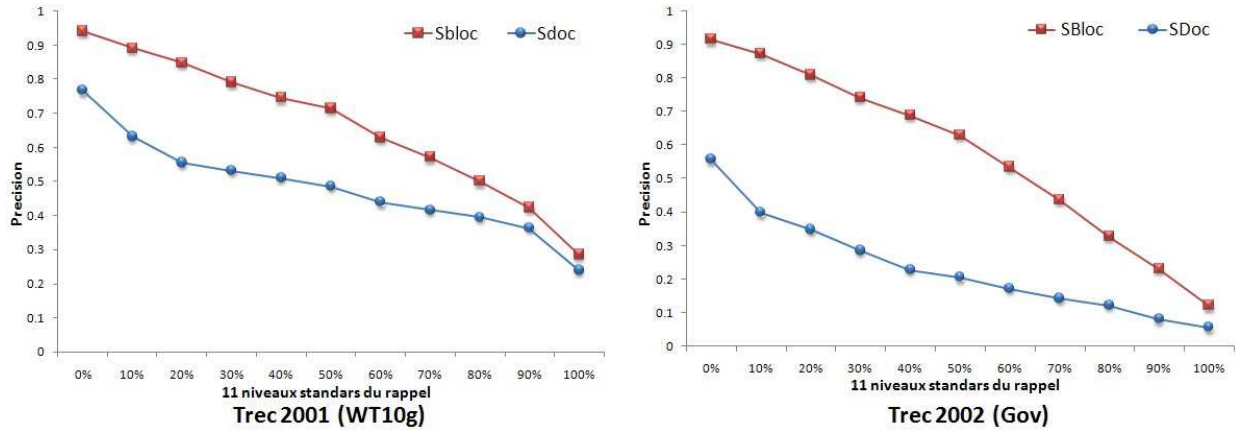
4.3 Expérimentations au niveau bloc thématique

Dans cette section, nous présentons les différentes expérimentations réalisées au niveau bloc. Dans un premier lieu, nous comparons deux fonctions de correspondance. La première repose sur les blocs thématiques que nous avons générés en utilisant un algorithme de segmentation décrit précédemment dans la partie modèle et la deuxième fonction de correspondance repose sur la page Web comme étant l'unité d'information la plus petite à retourner par un système de recherche d'information. Or, il n'existe pas de blocs pertinents aux différentes requêtes exécutées dans les jugements de pertinence des deux collections WT10g et GOV. Ce qui rend la comparaison entre les deux niveaux bloc et page difficile à effectuer. Afin de remédier à ce problème, nous avons calculé un score de pertinence d'une page P par rapport à la requête Q à partir des scores de pertinence des blocs qu'elle contient. Ce score est calculé comme suit :

$$\mathrm{Okapi}(\mathrm{P},\mathrm{Q}) = \sum_{\mathrm{B}\in\mathrm{P}} \mathrm{Okapi}(\mathrm{B},\mathrm{Q}) \tag{4.1}$$

Où Okapi(P,Q) et Okapi(B,Q) sont les scores de pertinence de la page P et du bloc B respectivement.
Dans un deuxième lieu, nous comparons notre fonction de voisinage appliqué aux blocs par rapport au contenu seul des blocs. Tout d'abord, on commence par la comparaison entre le niveau bloc et le niveau page.

La figure 4.4 montre les résultats expérimentaux obtenus sur la collection WT10g et GOV en utilisant deux fonctions de correspondance (bloc thématique et page). Notons SDoc l'algorithme qui calcule le score de pertinence des page en tenant compte du contenu seul de cette page et SBloc l'algorithme qui calcule un score de pertinence d'un page à partir des scores de pertinence reposant sur le contenu seul des blocs. D'après la



(a) (b)

FIGURE 4.4: La précision moyenne aux 11 niveaux standards du rappel du niveau bloc et page pour les deux collections WT10g (a) et GOV (b)

figure 4.4, nous remarquons que le calcul de pertinence au niveau bloc montre de meilleurs performances par rapport au calcul de la pertinence au niveau page. De plus, l'augmentation de la precision à tous les niveaux standards du rappel est très importante pour les deux collections WT10g et GOV. En effet, pour le niveau 0% du rappel, les résultats obtenus au niveaux bloc montrent une amélioration de 23% et 65% pour les deux collection WT10g et GOV respectivement par rapport au niveau page (la précision au 0% du rappel pour les deux collections WT10g et GOV est de 0,55 et 0,76 respectivement). L'amélioration de la précision aux autres niveaux standards du rappel de l'algorithme SBloc par rapport à l'algorithme Sdoc varie entre 16% et 53% pour la collection WT10g et de 115% à 210% pour la collection GOV. L'augmentation la plus importante et la plus significative concerne la collection GOV. Ces bons résultats obtenus au niveau bloc est dû à l'efficacité de notre algorithme de segmentation de pages Web en blocs thématiques que nous avons proposé et la capacité de notre système à cibler les informations pertinentes à la recherche demandée. De plus, le découpage des pages en blocs thématiques réduit considérablement les divergences existantes entre les documents par rapport à leur tailles. Dans ce qui suit nous allons voir le gain de la precision pour chaque requête exécutée.

La figure 4.5 montre le gain de la precision en MAP, P5 et P10 de l'algorithme SBloc reposant sur les blocs thématiques dans le calcul de pertinence des pages par rapport à l'algorithme de base Sdoc reposant sur le contenu seul des pages. Les résultats obtenus en fonction de ces mesures d'évaluation sont significatifs et prometteurs pour l'amélioration des performances des moteurs de recherche. En effet, moins de 7% des requêtes exécutées sur les deux collections WT10g et GOV réalisent des dégradations de precision faibles comprises entre 10% et 20% pour les mesures d'evaluations MAP, P5 et P10. Tandis que 70% à 80% des requêtes exécutées au niveau bloc montrent des améliorations significatives en la precision MAP, P5 et P10 comprise entre 20% à 100% par rapport au niveau page.

Ces résultats prouvent que le calcul de la pertinence au niveau bloc reste le meilleur moyen pour retrouver l'information recherchée et que notre algorithme de segmentation est adapté à tous les types de requêtes. Effectivement, le fait de découper une page en bloc permet à un moteur de recherche de ne retourner que l'information pertinente à la recherche qui se trouvait dans un bloc ou plusieurs blocs de la page. Le tableau 4.5 montre les résultats obtenus de la precision moyenne MAP, P5 et P10 sur les deux collections WT10g et GOV. Ces résultats confirment la performance de notre système basé sur le calcul de la pertinence au niveau bloc thématique par rapport à un système standard reposant sur le calcul de pertinence au niveau page.

De plus, au niveau bloc, il y a plus de documents pertinents au top du classement qu'au niveau page. Par exemple, sur la collection WT10g, les résultats montrent une amélioration de 43%, 60% et 55% sur la précision

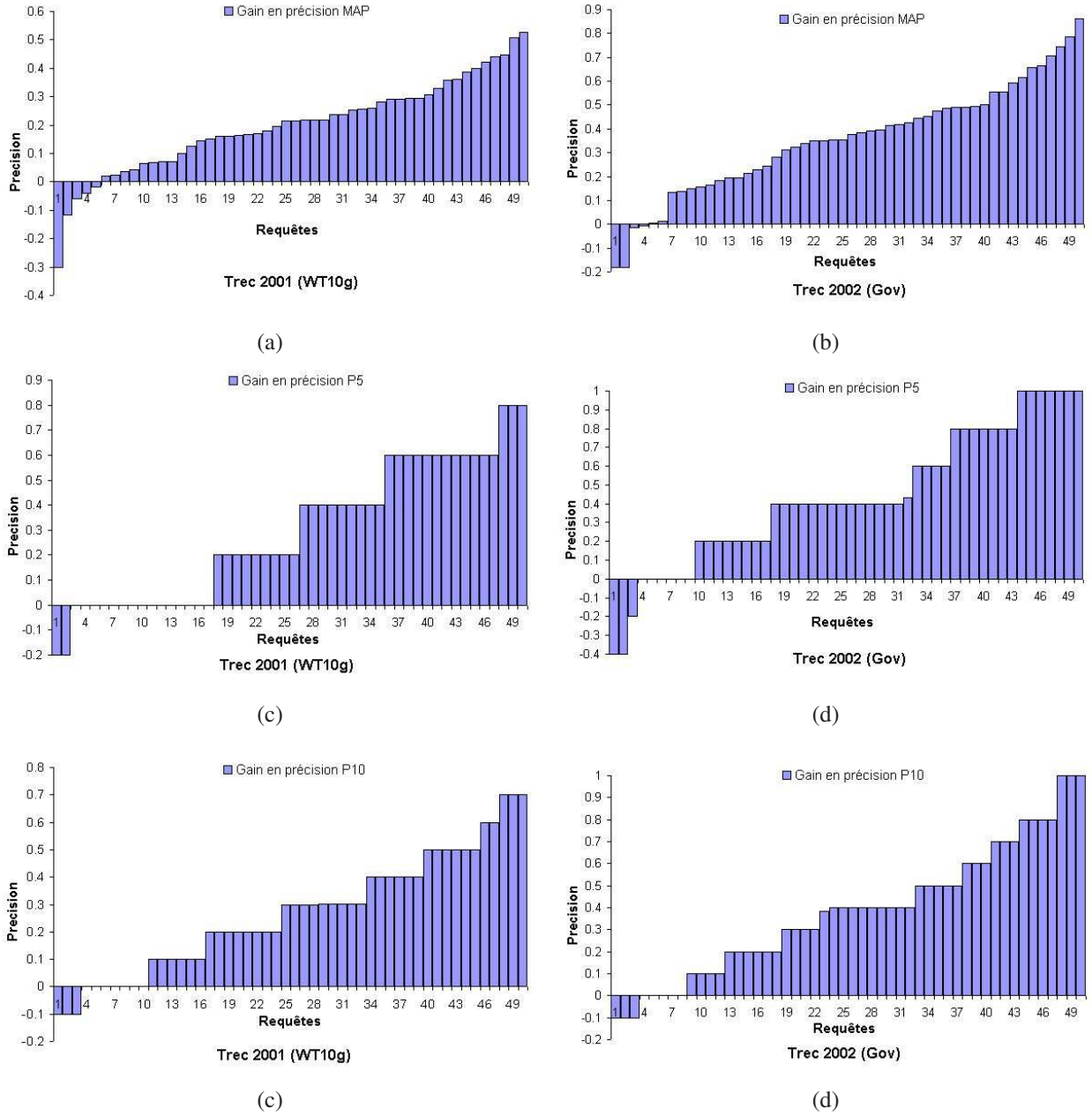


FIGURE 4.5: Gain de la précision en MAP, P5 et P10 du niveau bloc par rapport au niveau page

globale moyenne MAP, la précision moyenne P5 et P10 respectivement par rapport à l'algorithme de base SDoc. Nous avons le même constat sur la collection GOV dont les résultats montrent une amélioration importante de 171%, 153% et 146% sur la precision globale moyenne, la precision moyenne P5 et P10 respectivement par rapport à l'algorithme de base reposant sur le contenu seul des pages (SDoc). L'une des particularités de la collection GOV par rapport à la collection WT10g est la taille de ces documents en nombre de termes. Nous avons vu que la plupart des pages de la collection GOV sont volumineuses. Ceci explique pourquoi les résultats sont plus importants au niveau bloc. En effet, le poids du terme dans un bloc est calculé en fonction de la taille du bloc et de la fréquence de ce terme dans le bloc au lieu de la page entière. Il est fort possible avec la segmentation des pages que les blocs résultats ont des tailles raisonnables et que la densité des termes des requêtes dans ces blocs accroît la pertinence du bloc et la page qui contient ce bloc par rapport à la requête. Alors que, une dispersion des termes de la requête dans une page de grande taille augmente le poids de ces termes et de même le score de la page par rapport à la requête. Ce qui peut induire du bruit. La segmentation remédie au problème de l'influence de la taille du document dans le calcul du poids des termes que contient la page.

	Trec 2001(WT10g)			**Trec 2002(GOV)**		
Mesures	**SDoc**	**SBloc**	**Apport**	**SDoc**	**SBloc**	**Apport**
MAP	0.46	0.66	43%	0.21	0.57	171%
P5	0.5	0.8	60%	0.28	0.71	153%
P10	0.47	0.73	55%	0.26	0.64	146%

TABLE 4.5: Comparaison entre les deux algorithmes SDoc et SBloc en fonction de la precision moyenne MAP, P5 et P10

Enfin, en ce qui concerne la rapidité de retrouver les documents pertinents, illustré dans le tableau 4.6, le calcul de pertinence au niveau bloc reste plus performant que le calcul de pertinence au niveau page. En effet, le nombre de requêtes du système SBloc dont la premiere page retrouvée est pertinente à la requête posée enregiste une amélioration de 48% et 100% par rapport au système SDoc sur les deux collections WT10g et GOV respectivement. Le même constat est dressé par la mesure de succès à 5 et à 10 documents retrouvés (S@5 et S@10) dont les performances du calcul de pertinence au niveau bloc restent largement au dessus des performances du calcul de pertinence au niveau page avec des améliorations de 11% et 7% par rapport au mesures $S@5$ et $S@10$ respectivement sur la collection WT10g et de 53% et 28% par rapport aux mesures $S@5$ et $S@10$ respectivement sur la collection GOV. Par consequent, il est fort possible que les premieres pages retournées à l'utilisateur en repense à sa requête soient pertinentes à la recherche effectuée lorsque la pertinence de ces pages est calculée au niveau bloc au lieu du niveau page.

	Trec 2001(WT10g)			**Trec 2002(GOV)**		
	SDoc	**SBloc**	**Apport**	**SDoc**	**SBloc**	**Apport**
S@1	27	40	48%	21	42	100%
S@5	44	49	11%	30	46	53%
S@10	46	49	7%	36	46	28%

TABLE 4.6: Comparaison entre les deux algorithmes SDoc et SBloc par rapport au succès au 1^{er},5^{eme} et 10^{eme} documents retrouvés

4.4 Expérimentations des liens au niveau bloc

Dans cette section, nous allons voir l'impact de notre fonction de voisinage sur les blocs thématiques. Nous avons propagé les scores de pertinence des blocs dynamiquement à travers un graphe des blocs construit à partir du graphe des liens qui relie les pages de la même collection. Le processus de construction d'un tel graphe des blocs est détaillé dans la partie modèle de notre système. Nous avons appliqué le même procédé du calcul des scores de voisinage des pages au niveau bloc. Le tableau 4.7 montre les résultats obtenus en utilisant différentes variantes de notre fonction de propagation en fonction du paramètre de combinaison α. Nous distinguons trois scores différents : le score de propagation des poids de termes à travers le graphe des blocs (PPD_{imp}), le score de propagation des scores de pertinence des blocs à travers le graphe des blocs et enfin le score de propagation des poids des sites à travers le graphe de site. Nous avons combiné ces scores afin de calculer un score définitif d'une page par rapport à une requête utilisateur.

D'après le tableau 4.7, la combinaison entre une mesure reposant sur le contenu des blocs et le voisinage de ces blocs qui représente le score de propagation de pertinence calculé dynamiquement améliore les résultats obtenus sur la collection WT10g et GOV. Cette amélioration de la précision MAP et P5 varie entre 1% à 4% pour les deux collections WT10g et GOV. Nous remarquons que l'amélioration de la precision est faible. Ceci est justifié par une densité faible des liens dans le graphe des blocs. En effet, dans le calcul de voisinage des blocs, nous ne tenons compte que des blocs qui contiennent les termes de la requête. Or, il se peut que la projection d'un lien qui relie deux pages qui contiennent les termes de la requête dans le graphe des pages ne se traduise pas en un lien du graphe des blocs qui relie deux blocs qui contiennent les termes de la requête. Il suffit que le bloc source du lien ne contienne pas les termes de la requête pour que la propagation ne soit pas appliquée à ces blocs. De plus, il est fort possible qu'il existent beaucoup de liens dans les deux collections WT10g et GOV qui relient des blocs qui n'ont rien à voir avec la recherche effectuée (par exemple les liens publicitaires et les liens navigationnels. Nous remarquons aussi que la propagation des poids de termes et la propagation des scores de pertinence dynamique (PPD) améliorent la precision moyenne MAP et P5 du système reposant sur le contenu seul des blocs. Tandis que la propagation des poids des sites n'améliore pas assez les performances du système surtout quand α est grand. En effet, en donnant plus d'importance à l'information sur les sites, les performances du système se dégradent plus qu'elles l'améliorent. Ceci peut être justifier par le fait qu'un site Web contient plusieurs pages de thématiques différentes qui induisent du bruit dans les calculs des scores de pertinence. Enfin, l'importance des termes dans le calcul des poids de termes produit des améliorations sur les deux collections par rapport à l'algorithme reposant sur le contenu seul des blocs. Ces résultats montrent que la la combinaison entre l'importance local d'un terme à l'intérieur d'un bloc et son importance globale au sein de toute la collection ainsi que son importance par rapport au voisinage du bloc qui contient ce terme peut s'avérer utile et plus efficace dans la pratique par rapport à une pondération de termes standard en tenant compte de de l'importance locale et globale du terme. En conclusion, le modèle de propagation de pertinence que nous avons proposé peut s'adapter à n'importe quel niveau de granularité d'information.

α	MAP											
	Trec 2001 (WT10g)						Trec 2002 (GOV)					
	Sans PPT			Avec PPT			Sans PPT			Avec PPT		
	SD	**PPD**	**PPD+VS**	**SDimp**	**PPDimp**	**PPDimp +VSimp**	**SD**	**PPD**	**PPD+VS**	**SDimp**	**PPDimp**	**PPDimp +VSimp**
0	65.8%	65.8%	65.8%	66.2%	66.2%	66.2%	56.89%	56.89%	56.89%	57.2%	57.2%	57.2%
0.25	65.8%	66.08%	65.98%	66.2%	66.27%	66.34%	56.89%	57.22%	57.6%	57.2%	57.25%	57.56%
0.5	65.8%	66.43%	66.16%	66.2%	66.45%	66.73%	56.89%	57.54%	57.81%	57.2%	57.57%	57.88%
0.75	65.8%	66.61%	66.29%	66.2%	66.46%	66.79%	56.89%	57.71%	57.7%	57.2%	57.82%	57.8%
1	65.8%	66.53%	66.38%	66.2%	66.96%	66.7%	56.89%	58.14%	57.47%	57.2%	58.26%	57.74%
1.25	65.8%	66.51%	66.48%	66.2%	66.87%	66.68%	56.89%	58.41%	57.03%	57.2%	58.51%	57.44%
1.5	65.8%	66.46%	66.55%	66.2%	66.85%	66.68%	56.89%	58.48%	56.58%	57.2%	58.64%	57.03%
1.75	65.8%	66.48%	66.58%	66.2%	66.77%	66.62%	56.89%	58.39%	55.92%	57.2%	58.67%	56.46%
2	65.8%	66.55%	66.5%	66.2%	66.72%	66.63%	56.89%	58.43%	55.5%	57.2%	58.71%	55.97%
MAX	**65.8%**	**66.61%**	**66.58%**	**66.2%**	**66.96%**	**66.79%**	**56.89%**	**58.48%**	**57.81%**	**57.2%**	**58.71%**	**57.88%**

α	P5											
	Trec 2001 (WT10g)						Trec 2002 (GOV)					
	Sans PPT			Avec PPT			Sans PPT			Avec PPT		
	SD	**PPD**	**PPD+VS**	**SDimp**	**PPDimp**	**PPDimp +VSimp**	**SD**	**PPD**	**PPD+VS**	**SDimp**	**PPDimp**	**PPDimp +VSimp**
0	79.6%	79.6%	79.6%	80%	80%	80%	71.43%	71.43%	71.43%	71.84%	71.84%	71.84%
0.25	79.6%	80%	80%	80%	80.4%	80.8%	71.43%	71.84%	71.86%	71.84%	71.84%	71.43%
0.5	79.6%	80%	80.8%	80%	81.2%	82.4%	71.43%	71.84%	70.61%	71.84%	72.24%	71.02%
0.75	79.6%	80%	81.2%	80%	81.6%	82.4%	71.43%	72.24%	71.02%	71.84%	71.84%	72.24%	72.24%
1	79.6%	80.4%	81.2%	80%	82.4%	82%	71.43%	73.47%	71.02%	71.84%	73.06%	73.06%
1.25	79.6%	81.2%	81.6%	80%	82.4%	82%	71.43%	73.06%	71.43%	71.84%	73.06%	72.65%
1.5	79.6%	81.6%	81.6%	80%	82%	82%	71.43%	73.47%	71.02%	71.84%	73.47%	71.02%
1.75	79.6%	81.6%	81.6%	80%	82%	82%	71.43%	73.06%	69.8%	71.84%	73.06%	71.02%
2	79.6%	81.2%	81.6%	80%	81.6%	82%	71.43%	73.47%	69.8%	71.84%	73.47%	71.02%
MAX	**79.6%**	**81.6%**	**81.6%**	**80%**	**82.4%**	**82.4%**	**71.43%**	**73.47%**	**71.86%**	**71.84%**	**73.47%**	**73.06%**

TABLE 4.7: Comparaison entre différentes systèmes de recherche tenant compte des liens dans la fonction de correspondance au niveau bloc

Chapitre 5

Conclusion et perspectives

5.1 Apport de la thèse

Dans ce travail, nous avons abordé une problématique au centre des préoccupations actuelles des systèmes de recherche d'information qui est l'amélioration de la précision des réponses retournées par un moteur de recherche à un besoin utilisateur. Plusieurs travaux ont été menés sur l'utilisation des liens dans la recherche d'information sur le WEB mais, jusqu'à maintenant de nombreuses expériences ont montré qu'il n'y a pas de gain significatif par rapport aux méthodes de recherche basées seulement sur le contenu.

Au cours du chapitre 2, nous avons défini les principes généraux qui régissent les modèles de propagation en recherche d'information et répertorié les algorithmes les plus classiques. Nous avons voulu nous démarquer des autres états de l'art qui existent sur le sujet par un éclairage différent, avec notamment une classification de ces modèles selon plusieurs critères :

- dépendance du système par rapport à la requête (systèmes indépendants Vs systèmes dépendants)
- type de propagation (popularité ou pertinence)
- paramètre de propagation (fixe ou dynamique)
- granularité d'information considérée (blocs thématiques, pages, groupe de pages)

Nous avons soulevé certains problèmes majeurs de chaque approche étudiée dans l'état de l'art. Afin de palier les limites des modèles de propagation existants, nous avons proposé dans cette thèse un système de recherche d'information reposant sur un modèle de propagation de pertinence en utilisant à la fois le modèle Okapi et les liens hypertextes. Notre modèle de propagation de pertinence est inspiré des travaux de recherche présentés dans la section 2.3 du chapitre 2 qui portent sur la propagation de pertinence à travers le graphe du Web. La nouveauté dans notre système est l'utilisation d'une fonction de correspondance qui dépend du nombre de termes de la requête exécutée. En effet, la fonction de correspondance tient compte à la fois du contenu de la page et de son voisinage immédiat composé de pages qui la pointent. Ce voisinage est calculé dynamiquement en pondérant les liens hypertextes reliant les pages Web en fonction du nombre de termes de la requête contenus dans ces pages. Les fractions de scores de pertinence propagées à travers le graphe du Web sont proportionnels au nombre de termes de la requête contenus dans ces documents.

De plus, nous avons proposé une méthode de segmentation thématique qui nous permet d'extraire des blocs thématiques à partir des pages Web. Cette méthode de segmentation repose sur l'évaluation de plusieurs solutions de segmentation générées en utilisant des critères visuels et de représentations du contenu de la page HTML telles que les balises <HR>, <H1>..<H6>, <P>, <BR>, <TABLE>, etc. Cet algorithme est inspiré des algorithmes génétiques qui cherchent à retrouver une solution optimale à un problème donné parmi un ensemble de solutions heuristiques. Nous avons mis en evidence une analogie entre le problème de segmentation

de pages Web et la recherche d'un optimum dans un espace de solutions. Concernant l'implementation de cet algorithme, nous n'avons tenu compte que de la première itération de l'algorithme qui consiste à choisir la meilleure solution de segmentation parmi un ensemble de solutions de segmentation dont chacune d'elle est générée en utilisant un seul délimiteur de segments. Malheureusement, nous n'avons pas appliqué les opérateurs génériques (la mutation et le croisement) vu le nombre important de pages à segmenter. La fonction d'évaluation d'une solution de segmentation qui calcule la valeur d'adaptation d'une solution à notre problématique de segmentation repose sur deux mesures : la première repose sur la cooccurrence des termes qui permet de maximiser la cohérence entre les termes d'un même segment et l'autre mesure repose sur le calcul de la dissimilarité entre les blocs adjacents.

Nous avons calculé une valeur de pertinence d'un document Web à plusieurs niveaux d'abstraction (niveau bloc, page et site) en tenant compte de notre modèle de propagation de pertinence. Nous avons aussi proposé une architecture sur laquelle repose notre modèle de propagation.

5.2 Commentaires sur les résultats et prototype réalisé

Pour chaque niveau, nous avons réalisé des expérimentations : les premiers tests, concernant l'évaluation de notre modèle de propagation dynamique de pertinence par rapport à d'autres modèles existants reposant sur la propagation de popularité (PageRank [BP98] et HITS [Kle99]) et la propagation statique de pertinence (Savoy et al. [SR00]) ainsi que l'algorithme de base reposant sur le contenu seul des documents, ont été réalisés sur deux collections de documents WT10g et Gov avec deux ensembles de 50 requêtes différentes (TOPIC2001 et TOPIC2002). Plusieurs mesures d'évaluation des systèmes de recherche d'information telles que la précision moyenne globale (MAP), la précision à X documents retrouvés (P@5 et P@10), la précision aux 11 niveaux standards du rappel (0%, 10%, .. ,100%) et succès à X documents retrouvés (S@1, S@5 et S@10) ont été prises en compte pour évaluer les différents systèmes selon plusieurs critères (rapidité de retrouver les documents pertinents, précision en haut du classement, classement des documents les plus pertinents). La granularité de l'information considérée dans ces premières expérimentations est basée sur la page Web. Nous avons aussi évalué l'apport de l'importance des termes dans le calcul des poids des termes de l'index et le calcul de pertinence au niveau site sur les performances de notre système. Les deuxièmes tests concernent l'évaluation de notre approche qui repose sur le calcul de la pertinence des documents au niveau bloc au lieu de la page elle-même en utilisant l'algorithme de segmentation que nous avons proposé avec l'algorithme de base. Nous les avons comparés aux différentes mesures d'évaluation exposées ci-dessus. Enfin, nous avons évalué l'apport des liens au niveau blocs en utilisant le modèle de propagation dynamique de pertinence que nous avons proposé. De plus, nous avons évalué l'apport de l'importance des termes, et de l'information provenant des sites sur les performances de notre système.

Ces expérimentations ont confirmé la validité de nos approches. Notamment, nous avons montré, grâce à notre algorithme de segmentation thématique de pages, que le fait de calculer la pertinence d'un document par rapport à une requête utilisateur au niveau de granularité d'information plus petite que la page (blocs thématiques) pouvait conduire, outre le ciblage de l'information pertinente à l'utilisateur, à des résultats de meilleure qualité. Les résultats obtenus au niveau blocs montrent des performances considérables du système par rapport à d'autres systèmes reposant sur le calcul de la pertinence au niveau page. Cette amélioration varie entre 20% et 200% selon la mesure d'évaluation utilisée. En revanche, l'amélioration de notre système en utilisant les liens hypertextes n'est pas globalement très significative. Cependant, nous constatons que la plupart des documents pertinents à la requête utilisateur figurent dans le top du classement. Le modèle de propagation de pertinence reposant sur un paramètre de propagation calculé dynamiquement améliore le classement des documents pertinents à la requête, surtout dans les premières pages retournées par un système de recherche d'information classique. Nous avons remarqué que notre modèle de propagation de pertinence est adapté aux

requêtes dites génériques dont les termes figurent dans la partie haute de la page, par exemple, le titre et l'introduction de la page. C'est le cas des requêtes qui cherchent à retrouver les pages qui portent par exemple sur des endroits, des personnalités, des organisations, etc. En effet, il est fort possible que le texte ancre et le texte autour du lien contiennent des termes qui décrivent les pages pointées par les termes contenus dans les titres des pages pointées ou dans la partie haute de la page. De ce fait, notre modèle qui repose sur le nombre de termes de la requête du voisinage d'un document dans le calcul de son importance peut s'avérer utile pour améliorer les performances de la recherche. Cependant, pour le cas des requêtes dites spécifiques dont la partie pertinente se trouve dans le contenu du document, les performances du système se dégradent par rapport à l'algorithme de base. Ces requêtes cherchent à retrouver des informations sur des événements passés, des réponses à des questions posées comme *"Quel est le film qui a obtenu la palme d'or en 2000 ?"*. Avec ces requêtes, les liens n'apportent pas d'information additionnelle à la recherche parce que les termes de ces requêtes ne figurent pas dans le texte ancre ni autour du texte ancre des liens. De ce fait, le contenu restera le seul facteur pour déterminer la pertinence d'un document par rapport à la requête utilisateur.

5.3 Limites

Dans notre étude, nous avons focalisé notre travail sur les techniques de propagation de popularité et de pertinence, nous pouvons étendre la comparaison de notre modèle par rapport à d'autres techniques de propagation, notamment la propagation d'information. Les limites de nos expérimentations résident dans l'utilisation de collections de tests standards qui ont des failles :

1. Elles ne sont pas totalement représentatives du Web actuel, lequel contient des ressources hétérogènes.
2. Il est difficile de comparer des systèmes de recherche d'information avec les mesures d'evaluation actuelles. En effet, les jugements binaires de pertinence dans les deux collections standards WT10g et Gov de TREC ne permettent pas de comparer des systèmes au top du classement. Par exemple, deux systèmes qui retrouvent le même nombre de documents pertinents parmi les 10 premiers documents retournés en réponse à une requête posée sont considérés comme des systèmes ayant des performances identiques. Or, ni le classement et ni le degré de pertinence des documents jugés pertinents ne sont pris en considération dans l'evaluation des deux systèmes. En réalité, il existe des documents qui sont plus pertinents, peu pertinents ou moins pertinents que d'autres documents. Le recours à la logique flou est indispensable pour avoir une meilleure interpretation des jugements de pertinence. Nous avons aussi remarqué que plus de 40% des jugements de pertinence n'ont pas de liens entrants (pages isolées) dans les deux collections WT10g et Gov. De ce fait, la propagation de pertinence que nous avons proposée ne pouvais pas améliorer leurs classements. C'est pour cette raison que l'amélioration des performances de notre modèle de propagation de pertinence n'est pas assez significative.
3. Notre modèle de propagation de pertinence et de segmentation des blocs pourrait s'avérer utile pour déterminer des ressources pertinentes non textuelles en réponse aux requêtes spécifiques. Ces collections ne permettent pas d'évaluer la précision au niveau des blocs puisque les jugements de pertinence sont des pages Web. Par consequent, nous nous pouvons pas comparer notre algorithme de segmentation par rapport aux différents algorithmes existants de découpage de pages Web.
4. Notre modèle repose sur la pondération des liens en fonction du nombre de termes de la requête contenu dans les documents sources des liens. Or, le nombre de termes des requêtes exécutées sur les deux collections ne dépasse pas 4 termes et dans la plupart de cas elles contiennent moins de deux termes. Ces observations ne favorisent pas notre modèle de propagation dynamique de pertinence.
5. Nous n'avons pas participé faute de temps aux campagnes TREC spécifiques au Web ni aux campagnes liées au passage retrieval.

Une autre limite de notre système réside dans la complexité des calculs de notre algorithme de segmentation. En effet, plusieurs itérations sont nécessaires afin de segmenter une seule page. Les resources en mémoires et en temps augmentent en fonction du nombre de solutions considérées dans chaque itération. Notre algorithme de segmentation converge rapidement vers la solution optimale lorsque l'espace des solutions devient grand. Or, la taille de chaque solution correspond à la taille du document à segmenter, d'où des ressources considérables en matière de la mémoire vive des calculateurs. De plus, les calculs de co-occurrences entre les termes afin de mesurer la cohérence à l'intérieur d'un bloc nécessitent plusieurs accès disques au fichier qui contient la co-occurrence entre les termes puisque ce fichiers est trop volumineux et ne se tient pas en entier dans la mémoire vive. Comme, le nombre de pages à segmenter est important, il est difficile d'aller jusqu'à la solution la plus proche de l'optimum. Fixer le nombre d'itération de l'algorithme de segmentation est la solution idéale pour le processus de segmentation.

5.4 Perspectives

Dans les perspectives de ce travail, nous suggérons :

1. Application de notre modèle de propagation de pertinence dans la recherche contextuelle en tenant compte de différents types de voisinage (terme, bloc, page, site)
2. Application de notre modèle à des corpus de documents non HTML, par exemple les documents XML. En effet, le fait qu'un document XML soit semi-structuré peut faciliter la détection des blocs.
3. Evaluation de l'algorithme de segmentation que nous avons proposé par rapport à d'autres algorithmes de segmentation existants.
4. Application de notre l'algorithme de segmentation de pages Web en tenant compte des opérateurs de l'algorithme génétique que nous avons proposés (mutation et croisement) et en plusieurs itérations afin de mieux découper les documents en blocs thématiques.
5. L'utilisation de notre modèle de segmentation thématique dans le cadre de la recherche d'information personnalisée pourrait s'avérer efficace.
6. La modélisation d'un surfeur aléatoire thématique où nous distinguons différents déplacements dans le graphe des blocs thématiques. Des déplacements en gardant la même thématique que le bloc du départ et les déplacements en changeant la thématique du bloc. Le but est d'assigner une valeur de probabilité thématique à chaque bloc d'une page Web et l'utiliser pour calculer plusieurs scores de pertinence pour chaque nœud du graphe par rapport à chaque thématique étudiée. La figure 5.1 montre les différents déplacements existants à l'intérieur d'un graphe de blocs thématiques. la premiere étape du modèle de surfer aléatoire est d'associer une thématique à chaque bloc physique. Ceci nécessite une étude approfondie de la collection de test afin de déterminer les différents thèmes abordés dans cette collection. Deux scénarios émergent : une classification thématique des termes qui nécessitent l'utilisation des distances entre termes ou recours à un thesaurus tel que WordNet ou bien le choix judicieux d'un ensemble de descripteurs les plus représentatifs d'un bloc pour former la thématique du bloc. La deuxième étape du modèle d'un surfeur aléatoire thématiques consiste à propagé des scores de pertinence thématiques à travers le graphe des blocs thématiques en tenant compte des différents déplacement. Nous distinguons les déplacement à l'intérieur d'un documents et les déplacements externes au document. A l'intérieur d'un document, chaque bloc est relié au bloc qui le précède et le bloc qui le suive (développement linéaire des thématiques abordées à l'intérieur d'un document) ainsi que les blocs du documents de même thématique que lui (sauts thématiques). Dans ce cas de figure, deux probabilités différentes sont considérées : la probabilité de naviguer à l'intérieur d'un bloc présenté par PL et la probabilité de suivre un bloc de même thématique à l'intérieur d'un document présenté par PTL. Cependant, entre les documents, on distingue

trois types de déplacements : déplacement entre deux blocs en suivant les liens entre documents qui contiennent ces blocs et en gardant la thématique du bloc source du lien, déplacement entre deux blocs suivant les liens entre documents qui contiennent ces blocs en changeant la thématique, déplacement aléatoire en saisissons l'URL d'un document quelconque et en visitant l'un des blocs de ce document peu importe sa thématique car on ne sait pas à l'avance quels sont les thématiques abordées dans ce document. Nous distinguons deux probabilités différentes de navigation à l'extérieur d'un document : PLINK qui représente la probabilité qu'un surfeur aléatoire suit le lien entre deux blocs (lien entre les documents qui contiennent les deux blocs) et PET la probabilité de suivre la thématique du bloc à l'extérieur du document. Voici les différents pondérations des déplacements de la figure 5.1 :

– P1-2 → P1-0 : La probabilité d'un saut local thématique vaut PL*PLT
– P1-2 → P1-1 : La probabilité d'un saut local aléatoire vaut PL*(1-PLT)
– P1-2 → P1-3 : La probabilité d'un saut local aléatoire vaut PL*(1-PLT)
– P1-2 → P2-0 : La probabilité d'un saut externe thématique vaut (1-PL)*PLINK*PET
– P1-2 → P2-3 : La probabilité d'un saut externe non thématique vaut (1-PL)*PLINK*(1-PET)
– P1-2 → P3-0 : La probabilité d'un saut externe aléatoire vaut (1-PL)*(1-PLINK)

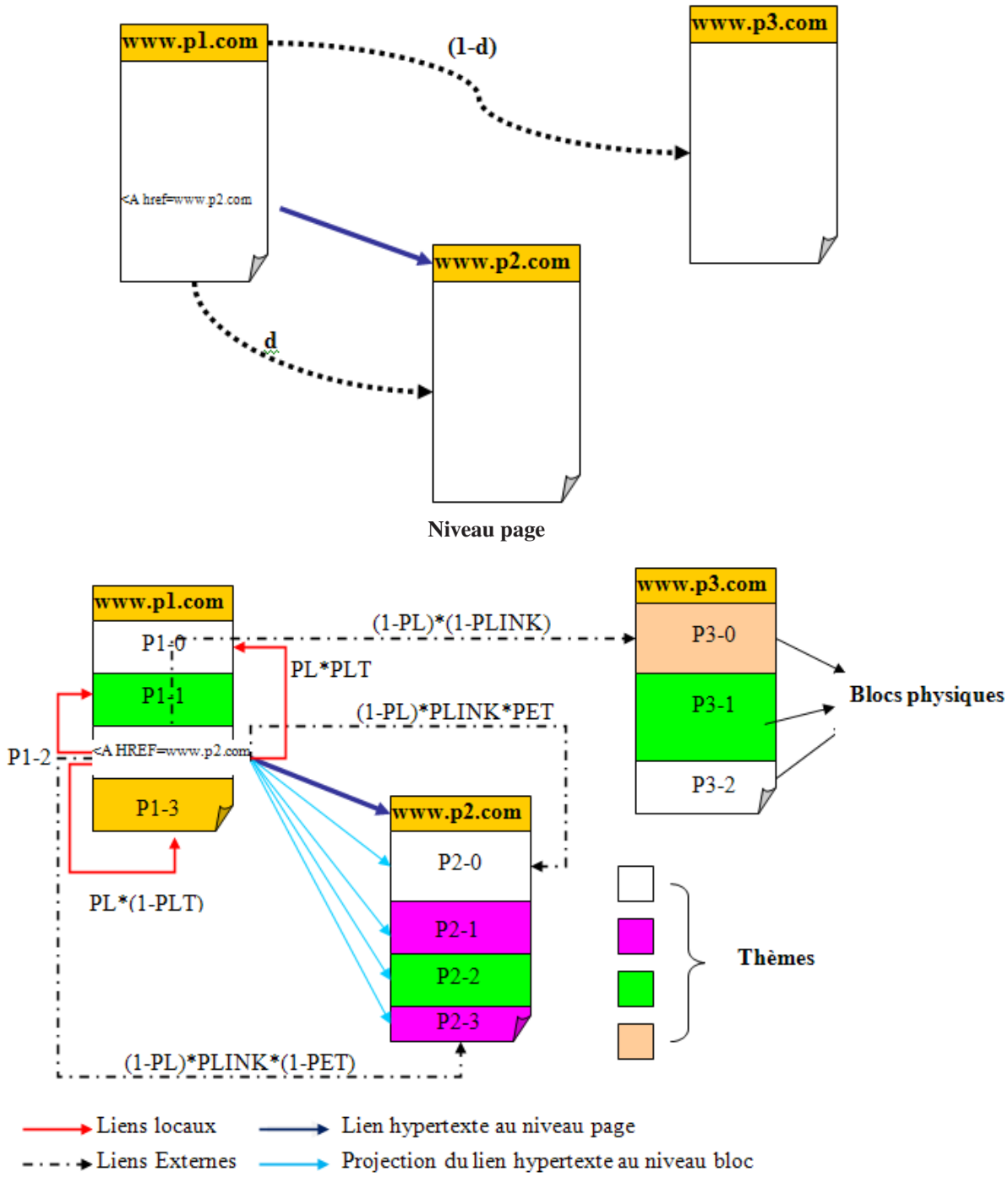


FIGURE 5.1: Modélisation d'un surfeur aléatoire thématique)

Chapitre 6

Annexe A : Méthodes d'évaluation des systèmes en recherche d'information

La qualité d'un système doit être mesurée en comparant les réponses du système avec les réponses idéales que l'utilisateur espère recevoir. Plus les réponses du système correspondent à celles que l'utilisateur espère, meilleur est le système. La démarche de validation en recherche d'information se base sur l'évaluation expérimentale des performances des modèles ou des systèmes proposés. Cette évaluation peut porter sur plusieurs critères : le temps de réponse, la pertinence, la qualité et la présentation des résultats, etc. Le critère le plus important est celui qui mesure la capacité du système à satisfaire le besoin d'information d'un utilisateur, c'est à dire la pertinence qui est une une notion complexe. Deux facteurs permettent d'évaluer ce critère. Le premier est le rappel, il mesure la capacité du système à sélectionner tous les documents pertinents. Le second est la précision, il mesure la capacité du système à sélectionner que les documents pertinents ou à rejeter tous les documents non pertinents. La mesure de précision et de rappel sont très utilisées sur des corpus textuels lorsqu'on connaît l'ensemble des éléments du corpus analysé. Cependant, ces mesures sont difficilement applicable dans le cas d'un moteur de recherche car il est difficile d'avoir une idée précise de l'ensemble des documents visibles sur le Web. Au fur et à mesure de l'évolution du domaine de recherche d'information, d'autres méthodes standard de mesure de qualité telle que la précision à X documents retrouvés ont été mises au point afin de pouvoir comparer aisément des algorithmes différents de RI.

L'évaluation des performances de la recherche d'information dans les premiers jours des systèmes de recherche d'information est focalisée principalement sur des expériences dans des laboratoires conçus pour cet objectif. Dans les années 90, beaucoup d'intérêt a été porté à l'évaluation des expériences dans le monde réel(sur le Web). En dépit de cette tendance, les expérimentations dans les laboratoires sont encore dominantes pour deux raisons principales : la répétitivité et l'échelle fournis dans ce cadre fermé de laboratoire. Une solution pour comparer plusieurs système de recherche d'information est le recours à une collection de tests composée d'un ensemble de documents, d'un ensemble de requêtes et des jugements de pertinence associés. Les jugements de pertinence sont des documents supposés être pertinents pour une requête donnée. La problématique réside dans la construction d'une bonne collection de test qui ressemble à celle du Web. Dans ce cadre, la communauté en recherche d'information s'est dotée de collections de tests. En effet, la conférence annuelle TREC (Text REtrieval Conference) est l'une des compagnes à offrir un corpus de documents et de requêtes très important afin d'évaluer les algorithmes de recherche d'information. Elle permet ainsi d'évaluer objectivement l'efficacité des algorithmes de recherche. Dans les sections suivantes, nous présentons les différentes mesures d'évaluations standards des systèmes de recherche.

Les mesures d'évaluation basées sur la notion de précision et de rappel comptent parmi les plus anciennes du domaine de la RI. La précision est le rapport du nombre de documents pertinents retrouvés sur le nombre

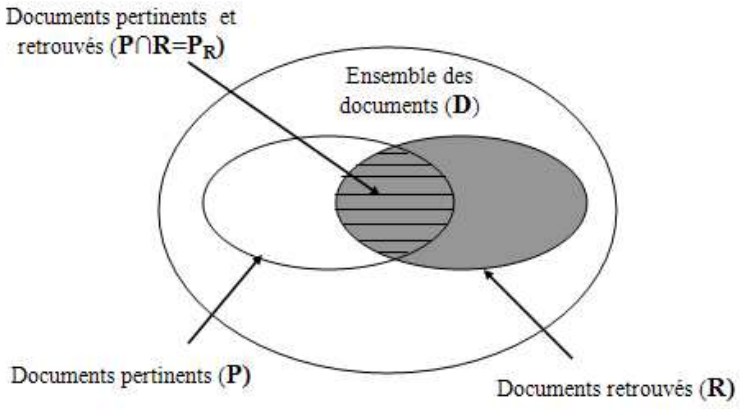


FIGURE 6.1: La précision et le rappel pour une requête donnée

total de documents retrouvés ; alors que le rappel est le rapport du nombre de documents pertinents retrouvés sur le nombre total de documents pertinents. À ces deux notions se rajoute une mesure d'efficacité globale, la F-measure initiée par Van Risbergen [Van Rijsbergen, 1979], qui représente la moyenne harmonique pondérée entre précision et rappel.

Considérons un exemple de besoin d'information et son ensemble P de documents pertinents. Soit $\mid P \mid$ le nombre de documents dans cet ensemble. Supposons une stratégie donnée de recherche d'information qui traite ce besoin d'information et produit un ensemble de réponse R. soit $\mid R \mid$ le nombre de documents de cet ensemble. De plus, soit $\mid P_R \mid$ le nombre de documents dans l'intersection des deux ensembles P et R. P_R est composé de documents pertinents au besoin d'information et retrouvés par la stratégie de recherche. La figure illustre ces différents ensembles.

Les mesures du rappel et de précision sont définies comme suit.

– Le rappel est la proportion de documents pertinents trouvés par rapport au nombre de documents pertinents.

$$Rappel = \frac{|P_R|}{|P|}$$

– La précision est la proportion de documents pertinents trouvés par rapport au nombre de documents trouvés.

$$Prcision = \frac{|P_R|}{|R|}$$

Les mesures dérivées du rappel et de la précision sont :

– Le bruit qui correspond aux résultats non pertinents trouvés par le système.

$$Bruit = 1 - P$$

– Le silence qui correspond aux résultats pertinents non trouvés par le système.

$$Silence = 1 - R$$

Remarque : Un rappel vaut 1 signifie que tous les documents pertinents sont retrouvés par le système. C'est à dire que l'ensemble des documents pertinents est inclus dans l'ensemble des documents trouvés ($P \subset R$) et non pas que seuls les documents pertinents soient retrouvés ($P = R$). De la même manière, une précision vaut 1 signifie que tous les documents retrouvés sont pertinents ($R \subset P$).

Le rappel et la précision, comme définis précédemment, supposent que tous les documents dans l'ensemble des documents trouvés (R) sont examinés ou vus. Cependant, l'utilisateur n'obtient pas tous les documents

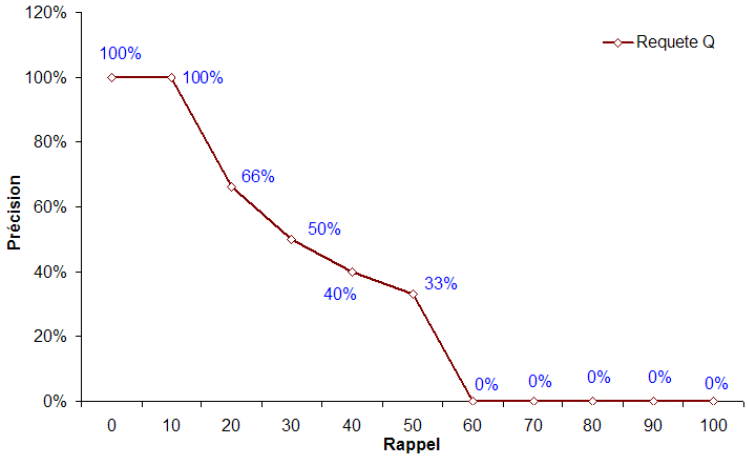


FIGURE 6.2: La courbe de la précision en fonction du rappel

à fois. Au lieu de renvoyer tous les documents trouvés d'un seul coup, les documents dans (R) sont d'abord ordonnés selon un degré de pertinence. L'utilisateur examine alors cette liste ordonnée de documents. Dans cette situation, les mesures, le rappel et de la précision, changent pendant que l'utilisateur poursuit l'examen de l'ensemble des résultats (R) .Ainsi, l'évaluation appropriée exige de tracer la courbe de précision en fonction du rappel comme suit :

Considérons une collection de documents et un ensemble de besoins d'information (ensemble de requêtes). Concentrons-nous sur un exemple donné d'un besoin d'information exprimé à travers une requête Q. Soit (P) l'ensemble contenant les documents pertinents pour la requête Q. Supposons que cet ensemble (P) est composé des documents suivants : $P = \{d3, d5, d9, d25, d39, d44, d56, d71, d89, d123\}$ Donc, selon un groupe de spécialistes, il y a dix documents qui sont pertinents à la requête Q. Considérons maintenant un nouvel algorithme de recherche. Et Supposons que cet algorithme renvoie une liste ordonnée de documents répondant à la requête Q. Cette liste est définie comme suit :

Classement	Documents trouvés	Pertinent/Non pertinent
1	d123	Pertinent
2	d89	Non pertinent
3	d56	Pertinent
4	d6	Non pertinent
5	d8	Non pertinent
6	d9	Pertinent
7	d511	Non pertinent
8	d129	Non pertinent
9	d187	Non pertinent
10	d25	Pertinent
11	d38	Non pertinent
12	d48	Non pertinent
13	d250	Non pertinent
14	d113	Non pertinent
15	d3	Pertinent

Si nous examinons la liste des documents retrouvés en commençant par les documents de haut du classement, nous observons les points suivants. D'abord, le document $d123$ classé numéro 1 est pertinent. De plus, il

correspond à 10% de documents pertinents à la requête Q. Ainsi, nous disons que nous avons une précision de 100% au rappel de 10%. En second lieu, le document $d56$ classé numéro 3 est le prochain document pertinent dans l'ensemble des document trouvés (R). A ce moment, nous disons que nous avons une précision approximative de 66%(deux documents pertinents sur trois trouvés) au rappel 20%(deux documents pertinents ont été visités sur les dix documents pertinents). Troisièmement, si nous procédons avec l'examen du classement généré, on aura la courbe de précision en fonction du rappel comme illustré dans la figure 6.2. La précision aux niveaux du rappel élevés (plus que 50%) vaut 0, parce que tous les documents pertinents n'ont pas été retrouvés. La courbe de la précision en fonction du rappel est habituellement basée sur 11 (à la place de dix) niveaux standards du rappel qui sont $0\%, 10\%, 20\%....100\%$. Pour le rappel de 0%, la précision est obtenue par un procédé d'interpolation comme détaillé au-dessous.

Dans l'exemple ci-dessus, les valeurs de la précision et du rappel se rapportent à une seule requête. Cependant, les algorithmes de recherche d'information sont généralement évalués par rapport à un ensembles de requêtes distinctes. Dans ce cas de figure, pour chaque requête, une courbe de précision en fonction du rappel est générée. Pour évaluer les performances de la recherche d'un algorithme par rapport à plusieurs requêtes différentes, nous calculons la précision moyenne pour chaque niveau du rappel de la manière suivante :

$$\overline{P}(r) = \sum_{i=1}^{N_q} \frac{p_i(r)}{N_q}$$

Où $\overline{P}(r)$ est la précision moyenne au niveau du rappel r, N représente le nombre de requêtes exécutées par le système, et $P_i(r)$ est la précision au niveau du rappel r pour la i^{me} requête (Q_i). Comme tous les niveaux du rappel pour chaque requête pourraient être distincte des 11 niveaux standard du rappel, l'utilisation du procédé d'interpolation est souvent nécessaire.

Par exemple, considérez encore l'ensemble des 15 documents ordonnés présentés ci-dessus. Supposant que l'ensemble de documents pertinents à la requête Q a changé et maintenant donné par : P= d3, d56, d129 Dans ce cas de figure, le premier document pertinent à la requête Q dans l'ensemble des documents trouvés est le document d56 qui représente 33.3% du rappel ($\frac{1}{3}$), avec une précision de 33.3% (un tiers des documents pertinents ont été déjà visités). Le deuxième document pertinent est le $d129$ qui représente un rappel de 66.6% ($\frac{2}{3}$), avec une précision de 25% ($\frac{2}{8}$). Le troisième document pertinent est $d3$ qui représente un rappel de 100% (tout les documents pertinents sont retrouvés), avec une précision égale à 20%. Les valeurs de précision aux 11 niveaux standard du rappel sont interpolées comme suit :

Soit rj, $j \in 0, 1, 2, .., 10$, le j^{me} niveau standard du rappel (i.e., r_5 représente le niveau 50% du rappel.

$$p(r_j) = \max_{r_j \leq r \leq r_{j+1}} p(r)$$

Dans notre dernier exemple, la règle d'interpolation produit les valeurs de précision illustrées dans la figure 6.3. Au niveaux 0%, 10%, 20% et 30% du rappel, la précision interpolée vaut 33.3% (qui représente aussi la précision connue au niveau 33.3% du rappel). Aux niveaux 40%, 50% et 60% du rappel, la précision interpolée est de 25% (qui correspond aussi à la précision au niveau $66,6\%$ du rappel). Aux niveaux 70%, 80%, 90% et 100% du rappel, la précision interpolée est de 20%(qui représente aussi la précision au niveau 100% du rappel).

Une approche additionnelle est de calculer la précision moyenne à X document retrouvés pertinents. Par exemple, nous pouvons calculer la précision moyenne aux 10^{me}, 20^{me}, ..100^{me} documents pertinents retrouvés. Le procédé est analogue au calcul de la précision moyenne à 11 niveaux standard du rappel mais fournit des informations additionnelles sur les performances des algorithmes de recherche.

Au fur et à mesure du développement du domaine de recherche d'information, d'autres mesures dérivées ont été mise en point pour comparer aisément différents systèmes selon différents critères de comparaison.

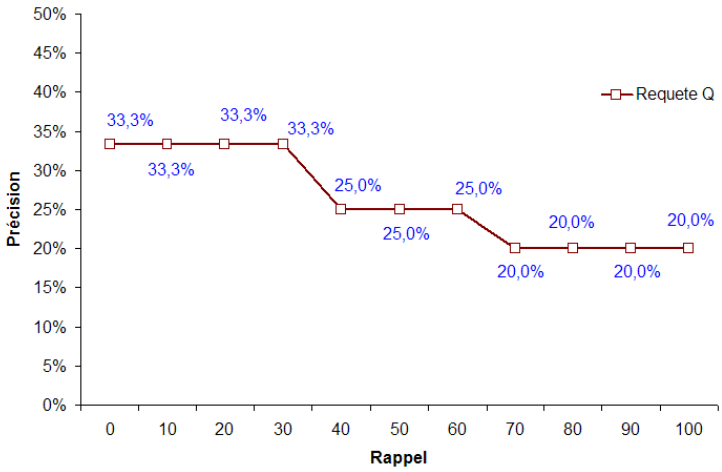


FIGURE 6.3: La courbe de la précision en fonction du rappel (cas d'interpolation)

Nous allons donner la signification et le but d'utiliser chaque mesure. Nous citons ces mesures que nous avons utilisées :

- **Précision moyenne MAP (Mean Average Precision) :** Elle repose sur la precision de chaque document pertinent trouvé. Elle est calculée pour chaque requête.
- **Précision moyenne aux** 11 **niveaux standard du rappel** 0%**,..,**100% **:** Cette mesure permet de voir les variations de la précision par rapport à des rappels différents. Le graphique de la precision moyenne aux 11 niveaux standards du rappel montre une courbe de la précision en fonction de 11 niveaux du rappel. Au lieu d'avoir que deux valeurs : la précision globale de la requête et son rappel, cette mesure calcule différentes précisions (au total 11).
- **Précision obtenue à X documents trouvés (P@5, P@10) :** Cette mesure permet de calculer la précision au top du classement, surtout la précision aux premiers documents retrouvés.
- **Succès au** 1^{er} **document trouvé (S@1) :** Cette mesure permet d'évaluer la rapidité de retrouver des documents pertinents. Elle est appliquée à un ensemble de requêtes pour voir le pourcentage des requêtes dont la première réponse est pertinente à la requête utilisateur.
- **Gain de la précision en MAP, P5 pour chaque requête exécutée :** Comme son nom l'indique, elle permet de voir la répartition du gain globale de la précision d'un système par rapport à un autre système sur les différentes requêtes utilisées.

Nos publications

[1] Idir Chibane and Bich-Liên Doan. Evaluation de la précision pour un système de recherche d'information hypertexte. In *Actes de la 2eme rencontres des sciences et technologies de l'information (ASTI'2005), Poster*, Clermont-Ferrand, France, October 2005.

[2] Idir Chibane and Bich-Liên Doan. Evaluation de la précision pour un système hypertexte. In *Actes de la 2eme Conference en Recherche d'Information et Applications (CORIA'05)*, pages 293–308, Grenoble, France, March 2005.

[3] Idir Chibane and Bich-Liên Doan. Precision evaluation of an information retrieval system using link and document scores. In *Proceeding of International Computer Systems and Information Technology Conference (IEEE ICSIT'2005)*, pages 253–258, Alger, Algeria, July 2005.

[4] Idir Chibane and Bich-Liên Doan. Relevance propagation model for large hypertext document collections. In *Proceedings of the IADIS International Conference WWW/Internet 2006 (IADIS'06)*, Murcia, Spain, October 2006. Actes sur CD-ROM, 8 pages.

[5] Idir Chibane and Bich-Liên Doan. Evaluation d'un modèle de propagation de pertinence dépendant des termes de la requête sur les collections wt10g et gov. In *Actes de Hypertext and Hypermedia : Products, Tools and Methods (H2PTM'07)*, pages 3–13, Hammamet, Tunisie, October 2007.

[6] Idir Chibane and Bich-Liên Doan. Impact of contextual information for hypertext documents retrieval. In *Proceedings of the 2nd International Workshop on Context-Based Information Retrieval (IWCBIR'07), in conjunction with the 6th International Conference on Modeling and using Context CIR*, pages 60–68, August 2007. ISSN 0109-9779.

[7] Idir Chibane and Bich-Liên Doan. Relevance propagation model for large hypertext. In *Documents Collections. Recherche d'Informations Assistée par Ordinateur (RIAO'07) : Large-Scale Semantic Access to Content (Text, Image, Video and Sound)*, Pittsburgh, USA, May 2007. Centre de hautes études internationales d'Informatique Documentaire (C.I.D.). Actes sur CD-ROM, 11 pages.

[8] Idir Chibane and Bich-Liên Doan. A web page topic segmentation algorithm based on visual criteria and content layout. In *Proceedings of the 30th Annual ACM Conference on Research and Development in Information Retrieval (SIGIR'07), Poster*, pages 817–818, July 2007.

[9] Idir Chibane and Bich-Liên Doan. Evaluation des performances d'un système de recherche d'information utilisant un algorithme de segmentation thématique de pages web. In *Actes de la 5eme Conference en Recherche d'Information et Applications (CORIA'08)*, pages 457–466, Trégastel, France, March 2008.

[10] Bich-Liên Doan and Idir Chibane. Expérimentations sur un modèle de recherche d'information utilisant les liens hypertextes des pages web. In *Actes des cinquièmes journées Extraction et Gestion des Connaissances (EGC'05)*, volume RNTI-E-3 of *Revue des Nouvelles Technologies de l'Information*, pages 257–262, Paris, France, January 2005. Cèpaduès-Editions.

Bibliographie

[Ade98] B. Adelberg. Nodose : a tool for semi-automatically extracting structured and semistructured data from text documents. *SIGMOD Record*, 27(2) :283–294, 1998.

[Agu99] I. Aguillo. Statistical indicators on the internet. *The European Science Technology Industry System in the World-Wide Web*, 1999.

[AK97] N. Ashish and C. A. Knoblock. Wrapper generation for semi-structured internet sources. *SIGMOD Record*, 26(4) :8–15, 1997.

[Ber02] M. L. Bernard. Criteria for optimal web design (designing for usability). *Usability News*, 2002.

[BGS05] M. Bianchini, M. Gori, and F. Scarselli. Inside pagerank. *ACM Trans. Inter. Tech.*, 5(1) :92–128, 2005.

[BH98] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *SIGIR '98 : Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 104–111, New York, NY, USA, 1998. ACM.

[BI01] L. Bjorneborn and P. Ingwersen. Perspectives of webometrics. *Scientometrics*, 50(1) :65–82, 2001.

[BI04] L. Bjorneborn and P. Ingwersen. Toward a basic framework for webometrics. *Journal of the American Society for Information Science and Technology*, 55(14) :1216–1227, 2004.

[Bjo01] L. Bjorneborn. Small-world linkage and co-linkage. In *HYPERTEXT '01 : Proceedings of the 12th ACM conference on Hypertext and Hypermedia*, pages 133–137, New York, NY, USA, 2001. ACM.

[BM02] K. Bharat and G. A. Mihaila. When experts agree : using non-affiliated experts to rank popular topics. *ACM Trans. Inf. Syst.*, 20(1) :47–58, 2002.

[BP98] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7) :107–117, 1998.

[Cal94] J. P. Callan. Passage-level evidence in document retrieval. In *SIGIR '94 : Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 302–310, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[CDKS02] S. Chien, C. Dwork, S. Kumar, and D. Sivakumar. Towards exploiting link evolution. In *Workshop on Algorithms and Models for the Web Graph*, November 2002.

[CH04] N. Craswell and D. Hawking. Overview of the TREC-2004 Web Track. In *Proceedings of TREC-2004*, Gaithersburg, Maryland USA, November 2004.

[Cha01] S. Chakrabarti. Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction. In *WWW'01 : Proceedings of the 10th international conference on World Wide Web*, pages 211–220, New York, NY, USA, 2001. ACM.

[CHWW03] N. Craswell, D. Hawking, R. Wilkinson, and M. Wu. Overview of the trec 2003 web track. In *TREC*, pages 78–92, 2003.

[CPS02] S. Chakrabarti, K. Punera, and M. Subramanyam. Accelerated focused crawling through online relevance feedback. In *WWW '02 : Proceedings of the 11th international conference on World Wide Web*, pages 148–159, New York, NY, USA, 2002. ACM.

[CYWM03] D. Cai, S. Yu, J. R. Wen, and W. Y. Ma. Vips : a vision-based page segmentation algorithm. Technical report, Microsoft Research, 2003.

[CYWM04] D. Cai, S. Yu, J. R. Wen, and W. Y. Ma. Block-based web search. In *SIGIR '04 : Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 456–463, New York, NY, USA, 2004. ACM Press.

[CZS+01] J. Chen, B. Zhou, J. Shi, H. Zhang, and Q. Fengwu. Function-based object model towards website adaptation. In *WWW '01 : Proceedings of the 10th international conference on World Wide Web*, pages 587–596, New York, NY, USA, 2001. ACM.

[DC05] B. L. Doan and I. Chibane. Expérimentations sur un modèle de recherche d'information utilisant les liens hypertextes des pages web. In *EGC*, pages 257–262, 2005.

[dSP63] D. J. d. S. Price. *Little Science, Big Science*. Columbia University Press,New York, 1963.

[Egg00] L. Egghe. New informetric aspects of the internet : some reflections, many problems. *Journal of information science*, 26(5) :329–335, 2000.

[EJN99] D. W. Embley, Y. Jiang, and Y. K. Ng. Record-boundary discovery in web documents. In *SIGMOD'99 : Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 467–478, New York, NY, USA, 1999. ACM.

[Fri87] M. E. Frisse. Searching for information in a hypertext medical handbook. In *HYPERTEXT '87 : Proceeding of the ACM conference on Hypertext*, pages 57–66, New York, NY, USA, 1987. ACM Press.

[FS92] H. P. Frei and D. Stieger. Making use of hypertext links when retrieving information. In *ECHT '92 : Proceedings of the ACM conference on Hypertext*, pages 102–111, New York, NY, USA, 1992. ACM Press.

[FS95] H. P. Frei and D. Stieger. The use of semantic links in hypertext information retrieval. *Inf. Process. Manage.*, 31(1) :1–13, 1995.

[Gar72] E. Garfield. Citation analysis as a tool in journal evaluation. science. *Science*, 178 :471–479, 1972.

[Gar83] E. Garfield. *Citation Indexing : Its Theory and Application in Science*. Technology and Humanities, The ISI Press, 1983.

[GCH+01a] J. Gao, G. Cao, H. He, M. Zhang, J. Y. Nie, S. Walker, and S. E. Robertson. Trec-10 web track experiments at msra. In *TREC*, 2001.

[GCH+01b] J. Gao, G. Cao, H. He, M. Zhang, J. Y. Nie, S. Walker, and S. E. Robertson. Trec-10 web track experiments at msra. In *Text REtrieval Conference*, 2001.

[GS05] A. Gulli and A. Signorini. The indexable web is more than 11.5 billion pages. In *WWW '05 : Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 902–903, New York, NY, USA, 2005. ACM Press.

[Hav02] T. H. Haveliwala. Topic-sensitive pagerank. In *WWW '02 : Proceedings of the 11th international conference on World Wide Web*, pages 517–526, New York, NY, USA, 2002. ACM Press.

[Haw00] D. Hawking. Overview of the trec-9 web track. In *TREC*, 2000.

[Hea94] M. A. Hearst. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 9–16, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[Hea97] M. A. Hearst. Texttiling : segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, 23(1) :33–64, 1997.

[HGmC+97] J. Hammer, H. Garcia-molina, J. Cho, R. Aranha, and A. Crespo. Extracting semistructured information from the web. In *In Proceedings of the Workshop on Management of Semistructured Data*, pages 18–25, 1997.

[HH76] M. A. K. Halliday and R. Hasan. *Cohesion in English (English Language)*. Longman Pub Group, 1976.

[HKJ03] T. Haveliwala, A. Kamvar, and G. Jeh. An analytical comparison of approaches to personalizing pagerank. Technical report, Université de Stanford, 2003.

[Hol62] J. H. Holland. Outline for a logical theory of adaptive systems. *Journal of the Association of Computing Machinery*, 3, 1962.

[Hol75] J. H. Holland. *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, 1975.

[Ing98] P. Ingwersen. The calculation of web impact factors. *Journal of Documentation*, 54(2) :236–243, 1998.

[JW03] G. Jeh and J. Widom. Scaling personalized web search. In *WWW '03 : Proceedings of the 12th international conference on World Wide Web*, pages 271–279, New York, NY, USA, 2003. ACM Press.

[JXS+04] X. M. Jiang, G. R. Xue, W. G. Song, H. J. Zeng, Z. Chen, and W. Y. Ma. Exploiting pagerank at different block level. In *WISE*, pages 241–252, 2004.

[Kes63] M. M. Kessler. Bibliographic coupling between scientific papers. *American Documentation*, 14(1) :10–25, January 1963.

[KHG03] S. D. Kamvar, T. H. Haveliwala, and G. H. Golub. Adaptive methods for the computation of pagerank. Technical report, Stanford University, 2003.

[KHMG03] S. Kamvar, T. Haveliwala, C. Manning, and G. Golub. Exploiting the block structure of the web for computing pagerank. Technical report, Stanford University, March 2003.

[KKM98] M. Y. Kan, J. L. Klavans, and K. R. Mckeown. Linear segmentation and segment significance. In *In Proceedings of the 6th International Workshop on Very Large Corpora*, pages 197–205, 1998.

[Kle99] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5) :604–632, 1999.

[KZ01] M. Kaszkiel and J. Zobel. Effective ranking with arbitrary passages. *J. Am. Soc. Inf. Sci. Technol.*, 52(4) :344–364, 2001.

[Lar96] R. Larson. Bibliometrics of the world wide web : an exploratory analysis of the intellectual structure of cyberspace. In *Annual Meeting of the American Society for Information Science*, Octobre 1996.

[Liu93] M. Liu. The complexities of citation practice : a review of citation studies. *Journal of Documentation*, 49(4) :370–408, 1993.

[LM00] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (salsa) and the tkc effect. *Comput. Networks*, 33(1-6) :387–401, 2000.

[LM02] A. N. Langville and C. D. Meyer. Updating the stationary vector of an irreducible markov chain. Technical report, North Carolina State University, Mathematics Department,CRSC, 2002.

[Mar73] I. V. Marshakova. A system of document connections based on references. *Scientific and Technical Information Serial of VINITI*, 6 :3–8, 1973.

[Mar97] M. Marchiori. The quest for correct information on the Web : Hyper search engines. *Computer Networks and ISDN Systems*, 29(8-13) :1225–1236, 1997.

[MH91] J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Comput. Linguist.*, 17(1) :21–48, 1991.

[Mol02] C. B. Moler. Cleve's corner : The world's largest matrix computation. Matlab news and notes, nst MATHWORKS, October 2002.

[NDQ06] L. Nie, B. D. Davison, and X. Qi. Topical link analysis for web search. In *SIGIR '06 : Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 91–98, New York, NY, USA, 2006. ACM Press.

[NZJ01a] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Link analysis, eigenvectors and stability. In *IJCAI-01 : Seventh International Joint Conference on Artificial Intelligence(IJCAI)*, pages 903–910, 2001.

[NZJ01b] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Stable algorithms for link analysis. In *Proceedings of the 24th Annual International ACM SIGIR Conference*. ACM, 2001.

[PBZ01] C. Prime, E. Bassecoulard, and M. Zitt. Co-citations and co-sitations : a cautionary view on an analogy. In *ISSI 2001 : Proceedings of the 8th International Conference on Scientometrics and Infometrics*, pages 529–540, New York, NY, USA, 2001. ACM Press.

[Pit99] J. E. Pitkow. Summary of www characterizations. *World Wide Web*, 2(1-2) :3–13, 1999.

[PN76] G. Pinski and F. Narin. Citation influence for journal aggregates of scientific publications : Theory, with applications to the literature of physics. *Information Processing and Management*, 12(5) :297–312, 1976.

[PND05] S. K. Pal, B. L. Narayan, and S. Dutta. A web surfer model incorporating topic continuity. *IEEE Transactions on Knowledge and Data Engineering*, 17(5) :726–729, 2005.

[Por80] M. F. Porter. An algorithm for suffix stripping. *Program*, 14 :130–137, 1980.

[Rey94] J. C. Reynar. An automatic method of finding topic boundaries. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 331–333, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[Rog77] P. M. Roget. *Roget's International Thesaurus (Fourth Edition)*. Harper and Row, 1977.

[Rou97] R. Rousseau. Citations : an exploratory study. *Cybermetrics*, 1(1), 1997.

[RSA97] K. Richmond, A. Smith, and E. Amitay. Detecting subject boundaries within text : A language independent statistical approach. In *In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 47–54, 1997.

[RWB+92] S. E. Robertson, S. Walker, M. H. Beaulieu, A. Gull, and M. Lau. Okapi at TREC. In *Text REtrieval Conference*, pages 21–30, 1992.

[Sma73] H. Small. Cocitation in the scientific literature : A new measure of the relationship between two documents. *Journal of the American Society for Information Science*, 24 :265–269, 1973.

[SR00] J. Savoy and Y. Rasolofo. Report on the trec-9 experiment : Link-based retrieval and distributed collections. In *TREC*, 2000.

[SSBM96] G. Salton, A. Singhal, C. Buckley, and M. Mitra. Automatic text decomposition using text segments and text themes. In *HYPERTEXT '96 : Proceedings of the the seventh ACM conference on Hypertext*, pages 53–65, New York, NY, USA, 1996. ACM.

[SWY75] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11) :613–620, 1975.

[SZ03] A. Shakery and C. X. Zhai. Relevance propagation for topic distillation uiuc trec 2003 web track experiments. In *TREC*, pages 673–677, 2003.

[SZ06] A. Shakery and C. X. Zhai. A probabilistic relevance propagation model for hypertext retrieval. In *CIKM '06 : Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 550–558, New York, NY, USA, 2006. ACM.

[TH03] M. Thelwall and G. Harries. The connection between the research of a university and counts of links to its web pages : an investigation based upon a classification of the relationships of pages to the research of the host university. *Journal of the American Society for Information Science and Technology*, 54(7) :594–602, 2003.

[TMS+03] A. C. Tsoi, G. Morini, F. Scarselli, M. Hagenbuchner, and M. Maggini. Adaptive ranking of web pages. In *WWW '03 : Proceedings of the 12th international conference on World Wide Web*, pages 356–365, New York, NY, USA, 2003. ACM Press.

[WM89] H. D. White and K. W. McCain. Bibliometrics. *Annual review of information science and technology*, 24(1) :119–186, 1989.

[WSC+03] J. R. Wen, R. Song, D. Cai, K. Zhu, S. Yu, S. Ye, and W. Y. Ma. Microsoft research asia at the web track of trec 2003. In *TREC*, pages 408–417, 2003.

[Yaa97] Y. Yaari. Segmentation of expository texts by hierarchical agglomerative clustering. In *In Proceedings of the Conference on Recent Advances in Natural Language Processing*, pages 59–65, 1997.